

Sitzungsprotokolle

Sitzungsprotokolle

Protokoll 19.12.11

Sensitivitätsanalyse

Abstract

Extended Management Summary

Einleitung

Anforderungen

Entwurf, 3D

Realisierung und Test

Schlussfolgerung

Video

Abnahme MS

Weiteres

Beschlüsse

Protokoll 16.12.11

Code Review

Beschlüsse

Protokoll 12.12.11

Abstract

Terminvereinbarung für Videoerstellung

Weiteres

Beschlüsse

Protokoll 09.12.11

Demo der Software

Weiteres

Beschlüsse

Protokoll 09.12.11

Review

Weiteres

Beschlüsse

Protokoll 05.12.11

Usability Test

Organisatorisches

Beschlüsse

Protokoll 28.11.11

Anforderungen-Dokument

Domain Analyse-Dokument

User Stories & Technische Probleme

Usability Test

Beschlüsse

Protokoll 25.11.11

Demo

User Stories

Weiteres

Protokoll 14.11.11

- [Protokolle](#)
 - [Präsentation der User Stories \(Sprint 4\)](#)
- [Protokoll 11.11.11](#)
 - [Planung Sprint 5](#)
 - [Externes Design](#)
 - [Termine](#)
 - [Beschlüsse](#)
- [Protokoll 07.11.11](#)
 - [Layout](#)
 - [Weiteres](#)
 - [Beschlüsse](#)
- [Protokoll 04.11.11](#)
 - [Code Review](#)
 - [Detail View und Filter entfernen](#)
 - [Weiteres](#)
- [Protokoll 31.10.2011](#)
 - [Verschieben auf 04.11.2011: Code Review mit Christian](#)
 - [Demonstration der für den Sprint 3 geplanten Must User Stories](#)
 - [Besprechung der Planung des Sprints 4](#)
 - [Prozessmodell](#)
 - [Weiteres](#)
 - [Beschlüsse](#)
- [Protokoll 24.10.2011](#)
 - [Notizen vom 17.10. besprechen](#)
 - [Protokolle vom 10.10. und 17.10 abnehmen](#)
 - [MS: Domain Model \(formelles OK Stolze \(UML\), inhaltliches OK Auftraggeber\)](#)
 - [Beschlüsse](#)
- [Protokoll 17.10.2011](#)
 - [Vision und Personas](#)
 - [Papierprototyp & Navigationkonzept](#)
 - [User Stories](#)
 - [Risikomanagement](#)
 - [Kommende Sitzungstermine](#)
 - [Beschlüsse](#)
- [Protokoll 10.10.2011](#)
 - [Creative Workshop](#)
 - [Planung letzter Sprint](#)
- [Protokoll 03.10.2011](#)
 - [Fragen Scrum](#)
 - [MS: Architekturprototyp \(Demo\)](#)
 - [MS: Vision Dokument V0](#)
 - [MS: Projektplan](#)
 - [MS: Benutzerbeobachtung/Befragung beendet](#)
 - [MS: Usability Tests](#)
 - [Weiteres](#)
 - [Beschlüsse](#)
- [Protokoll 26.09.2011](#)
 - [Bewertungskriterien](#)
 - [MS](#)

[Projektplanung](#)
[Agenda](#)
[Personas](#)
[Weiteres](#)
[Beschlüsse](#)
[Protokoll 19.09.2011](#)
[Dokumente, Deliverables](#)
[Termine](#)
[Vorgehen \(Scrum, RUP\)](#)
[Technologie](#)
[Resharper](#)
[Versionierungstool, Projektmanagementtool](#)
[Beschlüsse](#)

Protokoll 19.12.11

Erstellt am 19.12.2011, Autor: cheidt

Protokoll abgenommen am 20.12.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Sensitivitätsanalyse](#)
[Abstract](#)
[Extended Management Summary](#)
[Anforderungen](#)
[Entwurf, 3D](#)
[Realisierung und Test](#)
[Schlussfolgerung](#)
[Video](#)
[Abnahme MS](#)
[Weiteres](#)
[Beschlüsse](#)

Sensitivitätsanalyse

Agenda: Wie macht man eine Sensitivitätsanalyse für unsere Nutzwertanalyse? (ist die Vorhandene richtig gemacht?)

ist i. o. Keine Formeln und keine Statistik Angaben nötig.

Abstract

Fehlend: Deployment auf Surface 2 noch ausstehend

1. "Es gab bei Project Flip 1.0 Usability Probleme, um diese zu umgehen wurde ein Prototyp erstellt"

2. Intuitive Anwendung: "direkt bei der ersten Nutzen ohne Hilfe bedient werden kann"
3. "Zertifiziert" passt als Wort nicht (bei Usability Tests), besser "validiert"
4. "... um ein Konzept für die Gestaltung des User Interfaces zu definieren" - nicht Richtlinie
5. "Einen WPF Prototyp für die Perspective Wall Visualisierung entwickelt wurde." - es muss ersichtlich sein, dass etwas effektiv programmiert wurde - "...indem sie in Grundzügen eine Perspective Wall programmierte"

Extended Management Summary

Fehlend: Deployment auf Surface 2 noch ausstehend. Auch in Abstract erwähnen

Einleitung

i.o.

Anforderungen

Agenda: Themen: Kapitel 2 & 3
Muss nochmals geschickt werden.

Entwurf, 3D

Entwurf:

Bild von Surface 2 und nicht von Surface 1

Surface 2 als Box darstellen und die Umwandlung darin darstellen

Zudem kurze Erklärung oder Nummerierung

In Excel Datei wird spezifiziert wie der Einleseprozess von staten geht.

Architektur Entscheid: Viele Vorgaben, daher wenig Entscheidungsfreiheiten. Was gehört zu Designentscheiden?

Designentscheid:

- Warum Daten in Textfile und nicht in Datenbank?
- Verwendung von Verbindung zu Webserver (Connector) mit Interface

3D:

Kann ein eigenständiges Dokument oder ein Unterkapitel von Entwurf sein

Fokus und Kontext als Begriffe einführen. Weiter unten schon erste Verletzung dieser Regel (da nicht der gesamte Kontext angezeigt wird). In gewisserweise Verletzung des Prinzips.

Realisierung und Test

Schreiben, dass uns mittlerweile bewusst ist, dass es einen Standardfragebogen gibt. Wir haben diesen speziellen Fragebogen jedoch selber erarbeitet und daher benutzt. Mehr Bezug bei Fazit bei Usability Test auf die Auswertung?

Schlussfolgerung

stark überdeckt → abgedeckt?

Video

Agenda: MS: Review Draft Wiki Page und Video. Extended Management Summary auf Wiki? - ist so abgenommen. Am besten mit 1-2 Bildern.

Nicht zukünftige Lösung sondern entwickelte Lösung für Surface 2. Demonstration auf HP Touch Smart Bildschirm (siehe User Interfaces 2)

Letztes Zwischenbild Surface 2 Bild, Deployment vom Zühlke Engineering geplant.

Abschlussfolie mit Implementationstechnologien (WFP, Surface 2 Toolkit, Testing Tools etc.)

Abnahme MS

MS: Review Inhaltsverzeichnis Bericht - schicken

MS: Review der geplanten Aufwände der User Stories und Tasks für die nächste Iteration - i.o.

MS: Abschluss von Iterationen: Jedes Iterationsende wird von Stolze oder Auftraggeber abgenommen - i.o.

Weiteres

Zeitauswertung in richtiges Dokument und nicht in Anhang

Dokumentaufteilung nach eigenen Vorstellungen

Projektretroperspektive: Schätzung und Zeitaufwendung und Lessons Learned

Charts für Zeitauswertung: Was ist spannend, wo gab es Überraschungen

Beschlüsse

Folgende MS sind abgenommen:

- MS: Review Inhaltsverzeichnis Bericht (wird noch geschickt)
- MS: Review der geplanten Aufwände der User Stories und Tasks für die nächste Iteration
- MS: Abschluss von Iterationen: Jedes Iterationsende wird von Stolze oder Auftraggeber abgenommen
- MS: Review Draft Wiki Page und Video

Sitzungsprotokoll & Inhaltsverzeichnis schicken

Dokumente korrigieren

Protokoll 16.12.11

Erstellt am 16.12.2011, Autor: Ielmer

Protokoll abgenommen am 16.12.2011 durch Michael Gfeller

Anwesend:

Michael Gfeller, Lukas Elmer

Agenda:

[Code Review](#)

Code Review

- Laufzeit Warnings: Beim Binding gibt es Laufzeit Warnings und zwar in der Detailansicht beim Navigieren von einer Project Note zur nächsten. Dies wird durch das Binding verursacht und könnte daran liegen, dass auf null gebündelt wird. Die Behebung dieser Warnings würde wahrscheinlich zu viel Zeit in Anspruch nehmen.
- Im GravatarsViewModel wird eine IList verwendet. Besser wäre hier eine ListView, da durch das automatische Binding / Konvertieren ein Memory Leak entstehen könnte.
- Durch die Installation des WPF Performance Kit könnte man die Ressourcen besser überwachen und beobachten.
- GravatarsViewModelTest: Dieser Test bringt nur wenig Mehrwert.
- Die Konfiguration app.config in den unteren Projekten (Bsp. PdfConverter) bringt nichts, da die Datei nicht ins Verzeichnis mit den ausführbaren Dateien kopiert wird.

Beschlüsse

- Aufgrund Zeitdruck ist es nicht möglich, alle beanstandeten Punkte umzusetzen. Deshalb werden die Laufzeit Warnings, die sehr wahrscheinlich durch das Binding auf null Objekte ausgelöst wird, ignoriert.
- Die Konfiguration wird noch kopiert, damit Einstellungen vorgenommen werden können.
- Die anderen Punkte werden berücksichtigt, falls noch Zeit bleiben sollte.

Protokoll 12.12.11

Erstellt am 12.12.2011, Autor: cheidt

Protokoll abgenommen am 19.12.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Abstract](#)

[Terminvereinbarung für Videoerstellung](#)

[Weiteres](#)

[Beschlüsse](#)

Abstract

1. Abschnitt: höchstens noch an Details feilen (z.b. Im Bereich Clienttechnologien, Kunden die Möglichkeiten von Microsoft Surface als Ausstattungsgerät aufzeichnen).
2. Abschnitt: Für dieses Projekt wurde ein benutzerorientiertes Vorgehen gewählt, bei welchem die gefundenen Dinge umgesetzt wurden. Optimierung der einfach, intuitiven Anwendung, ohne Hilfsmittel war der Fokus dieser Arbeit.
3. Abschnitt: Prozesskomplexität wegen Benutzerzentrierung. Immer wieder Testen der User Interfaces, wie oft?
Client-Teil und Serveranbindung.
Obwohl für alle Teammitglieder WPF und .NET neu war, ist es gelungen einen vollständigen Prototypen zu entwickeln.
Integration von den Kategorien nennen (dynamisches Laden, Aggregationsdatei). Reduzierung der Komplexität.

Prototyp für Perspective Wall, keine Integration wegen mangelnder Zeit (vorgeschrittene Visualisierungskonzepte).

Terminvereinbarung für Videoerstellung

Multimedien dienst verleiht und reserviert Videokameras

Videos können auch konventioneller sein

- Wichtig sind sauberer Start- und Endscreen und Demo des Endsystems
- Abstract wird noch einmal erklärt
- Beispiel der bestehenden Lösung: Papierformat filmen
- Beispiel des Prototypen
- Prototyp Perspective Wall ebenfalls zeigen
- Am Ende Namen und Kontaktemail

Markus auflisten, mit HSR Adresse. Wir können auch private Adresse nehmen (wegen Ablauf der Emailadresse).

Weiteres

- Aufgabenstellung unterschreiben -> ok
- **Frage:** User Stories: was kommt in Anforderungsdokument (Definition of Done, Sprint, Abnahmedatum, Ticketnr)?
Antwort: Kann selbst entschieden werden.
Abnahme Tests werden mit den Anforderungen verglichen. Definition of Done muss dafür ersichtlich sein. -> User Stories mit Definition of Done und Sprint in Anforderungsdokument. Auf Excel im Anhang verweisen.
- **Frage:** ISO Norm Usability Test, Standardbogen übernehmen?
Antwort: Muss nicht angepasst werden
- **Frage:** Code Files mit Header information; Autor; Datum von Code Reviews
Antwort: Termine hätten definiert werden müssen für Code Reviews (von Anfang an). In Anforderungsdokument notieren, dass Code Reviews kontinuierlich und partnerschaftlich durchgeführt wurden und daher nicht im Code vermerkt wurden. "Die folgenden Codereviews wurden mit externen Personen durchgeführt" mit Auflistung. Autorenschaft ist gemeinschaftlich und wird daher nicht aufgelistet.
- **Frage:** In welches Dokument gehört die Beschreibung der Tools?
Antwort: Anforderungsdoku

Beschlüsse

- Abstract anpassen/erweitern
- User Stories mit Definition of Done und Sprint in Anforderungsdokument. Auf Excel im Anhang verweisen.
- Dokumentation der Code Reviews und der Tools in Anforderungsdokument

Protokoll 09.12.11

Erstellt am 09.12.2011, Autor: cheidt

Protokoll abgenommen am 12.12.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Christian Moser, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Demo der Software](#)

[Weiteres](#)

[Beschlüsse](#)

Demo der Software

Agenda: Demonstration der Software und Abnahme der User Stories

- Touch Target bei Vor- und Zurück-Buttons und Lupen vergrössern.
- Virtualisierung für zukünftige Weiterentwicklung
- Minimale Höhe für Scrollbar einstellen (muss nicht umgesetzt werden).
- Kundenliste hat kritische Höhe (viele Elemente) -> Grouping nach Alphabet
- Gedrückter Button sollte ebenfalls weiss sein bei Filter
- Filterkriterienliste wäre besser grösser (Ausnutzung des Bildschirms)
- InfoView:
 - Schriftart für i (serifen)
 - Kreuz für das Schliessen der Ansicht einfügen
 - Für Überschriften AA Zühlke Medium wählen und Schrift orange einfärben
 - Dropshadow für Bilder im InfoView
 - Bilder statisch für InfoView

Userstories wurden abgenommen

Weiteres

Abklärung von Christian:

- Termin für Präsentation Ende Februar
- Kann Projekt weitergezogen werden?

Installationsanweisung: Was braucht es für Verbindungen, um die Project Notes herunterzuladen

Nicht extrem ausformulieren, es muss funktional sein

Beschlüsse

User Stories wurden erfolgreich abgenommen

Protokoll 09.12.11

Erstellt am 09.12.2011, Autor: cheidt

Protokoll abgenommen am 09.12.2011 durch Michael Gfeller

Anwesend:

Michael Gfeller, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Review](#)

[Weiteres](#)

Review

- Styles in App.xaml positionieren oder Styles.xaml mit Styles erstellen und in App.xaml referenzieren. Einzelne Styles in auslagern und mittels Dictionary zusammen mergen (Merged Resource Dictionaries).
- Images nicht freezed → dadurch evt. Memory Leaks
- ScrollToTopBehavior: DependencyPropertyDescriptor ist statisches Konstrukt (ItemsSourceProperty): prüfen, dass es sich abmeldet
- Aufräumen von Properties, privaten Variablen und Methoden
- Absoluter Pfad zu Adobe Reader
- ViewModel darf nicht UserInterface zugreifen. Eigenes Projekt für ViewModel. Base könnte entfernt werden (da es nur ein ViewModel gibt). Aufteilung des ViewModels auf mehrere ViewModels nicht nötig. Bei einem View Model KEIN eigenes Projekt.
- Installer fehlt, da aber keine Hardware verfügbar ist, ist dies nicht nötig. In Dokumentation erklärt, was alles noch getan werden muss, um Projekt auf Surface 2 zu benutzen.
- Code dokumentieren für public, falls sich dies bei Abschnitt lohnt (wenn nicht klar ist, um was es sich handelt, z.B. bei Preload()).
- Resharper Warnings-Disabling-Kommentare in Tests entfernen, falls diese nicht interessant sind
- Wenn Styles in App.xaml definiert sind, dann werden diese auch mit Static Resource gefunden.

Weiteres

Lizenzen von dotTrace etc. weiterleiten

Beschlüsse

ViewModel: Da nur 1 ViewModel existiert, wird kein eigenes Projekt für die ViewModels erstellt.

Protokoll 05.12.11

Erstellt am 05.12.2011, Autor: cheidt

Protokoll abgenommen am 12.12.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Usability Test](#)
[Organisatorisches](#)
[Beschlüsse](#)

Usability Test

Agenda: mündlicher Kurzüberblick über Ergebnisse der Usability Tests
Verschoben auf kommenden Montag, 12.12.11

Organisatorisches

- Wo in Bericht Vereinbarung über Urheber- und Nutzungsrechte?
Können wir die Vereinbarung im Stil unserer Dokumente machen (zusätzlich HSR-Logo)?
Unterschriebene Dokumente kommen in den Anhang.
Dokument darf angepasst werden
Unterschrift Zühlke? -> noch einfügen
Unterschrift Studiengangleiter? -> Feld für Studiengangleiter kann weggelassen werden
Gemeinsames Nutzungsrecht für alle
- In welches Dokument gehört die Beschreibung der Tools? -> Gruppe Egli/Schnyder fragen, ansonsten in Architekturdokument (Software Engineering Umgebung)
- Redmine Wiki auf CD archivieren aber nicht in PDF einfügen
- Codereview ist unabhängig von Dokumentation bei der Bewertung der Arbeit
Termin vereinbaren mit Assistent zum Codereview. Codereview mit Michael Gfeller
- In welches Dokument gehört das Betriebskonzept? was muss wöchentlich, monatlich oder jährlich gemacht werden, damit das System weitergeführt werden kann.
- Platzhalter für Installation, der besagt, was der aktuelle Stand des Wissens ist. Bei Erneuerung müsste dieser Teil noch einmal neu überarbeitet werden.
- Dokument die die notwendigen Arbeiten auflisten, die wegen der fehlenden Hardware verschoben werden. Oder Softwareelemente, die zeitlich nicht umgesetzt werden konnten. Dokument für Weiterentwicklung
- Aufgabenstellung: Auch Unterschrift der Zühlke benötigt? -> Aufgabenstellung nur von Markus Stolze unterschreiben lassen
- Sitzung vom 9.12.: Demo der Software
- Zitierung: Autornamen + Jahr, bei mehreren Autoren, erster Autorenname + et al. Runde Klammern bei ganzem Namen
- Bei Zusammenfassung von Dokument zu einem PDF, das Titelblatt muss nicht entfernt werden.
- Gesamtinhaltsverzeichnis erstellen (wie?). Nur von gewissen Hauptüberschriften

Beschlüsse

Usability Test Überblick: Verschoben auf kommenden Montag, 12.12.11
Codereview: Termin vereinbaren (findet am 9.12.11 statt)

Protokoll 28.11.11

Erstellt am 28.11.2011, Autor: cheidt

Protokoll abgenommen am 05.12.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Anforderungen-Dokument](#)

[Domain Analyse-Dokument](#)

[User Stories & Technische Probleme](#)

[Usability Test](#)

Anforderungen-Dokument

- Funktionale Anforderungen fehlen (User Stories).
Liste von kurzen Namen der User Stories in Dokument hereinnehmen, Details dazu können in Excel-File bleiben, mit Verweis darauf.
- Angemessenheit: Die Angemessenheit wird in Form eines Usability Tests getestet und folgende Szenarien werden dabei getestet (Messform).
- Reinigung gehört zu Betriebskonzept, wird ein solches Dokument überhaupt erstellt?
Betriebskonzept als Rumpf (richtiges kommt zu einem späteren Zeitpunkt.
Dokumentieren was jetzt schon bekannt ist. Betriebskonzept müsste später noch ausgearbeitet werden).
- Attraktivität: Corporate Design wird genutzt (ist auch überprüfbar). Fernwirkung bei Attraktivität auflisten? Fern- und Nahwirkung zur Attraktivitätssteigerung. Animation bei Bildwechsel bei Nahwirkung.
- Änderbarkeit und Wartbarkeit zu einem Kapitel zusammen nehmen.
- Bei den Design Constraints gehört eine Referenz dazu, dass das Corporate Design von der Zühlke Engineering AG verwendet wurde (wie schon bei externen Design angesprochen).
- XPS-Konvertierung ist ein Design Entscheid und kein Design Constraint.

Dokument muss nach Korrektur nicht geschickt werden.

Domain Analyse-Dokument

Project Flip 1.0 befindet sich in Vorstudie. Referenz?

Domain Model: Nicht verständlich, was mit "Attribute derselben Oberkategorie" gemeint ist. Gut wäre ein Beispiel von einer Project Note.

Kategorien werden mit Begriffen vermischt → alles zu Kategorie unbennenen.

Creative Workshop: Mit wem und wann wurde dieser durchgeführt.

Dokument muss nach Korrektur nicht geschickt werden.

User Stories & Technische Probleme

Agenda: User Stories abnehmen

Zoom über durchsichtiges, darüberliegendes Element lösen für den Document Viewer. Etwas zweites Durchsichtiges um zu scrollen oder wieder auszuzoomen.

Optische Täuschung bei den Filterkriterien → Bearbeiten.

Projektdaten: bei Navigation wieder ganz nach oben scrollen.

Screenshot des Fensters bei Navigation

3-D Elementübersicht unbedingt dokumentieren

Usability Test

Agenda: Termin für Usability Test vereinbaren

Termin: 5.12.2011, 11:30-12:00

Beschlüsse

Anpassungen an Dokumenten vornehmen (Project Flip 1.0: Die Analyse von Project Flip 1.0 befindet sich im Dokument Vorstudie. Im Dokument Domain Analyse wird keine Referenz auf dieses Kapitel gesetzt.)

Protokoll 25.11.11

Erstellt am 25.11.2011, Autor: cheidt

Protokoll abgenommen am 20.12.2011 durch Christian Moser

Raum: 6.112, Gebäude 6 Zeit: 15:00

Anwesend:

Christian Moser, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Demo](#)

[User Stories](#)

[Weiteres](#)

Demo

Agenda: Demo der Applikation, Fragen

Demo:

- Dialog schliessen wenn auf gleiches Element geklickt wurde.
- Hintergrund etwas weniger abdunkeln bei Menuauswahl. Ist Verdunkelung überhaupt notwendig?
- Texttrimming: Dokpanel anstatt Stackpanel bei Project Note oder bei Text feste Feldgrösse.
- Farbe bei "Kriterien hinzufügen" auf weiss wechseln (mehr Kontrast).

Fragen:

- Mit Helper arbeiten (statische Klasse).

- Arbeiten mit Code Behind i.o., solange diese Teile nicht getestet werden müssen.
- 3-D Effekt nicht weiter verfolgen, da Aufwand zu gross.

Event bei Scrollen, inkl. Timer, nach 1-2 Sekunden wird er wieder abgestellt, mit Property welches gebunden wird.

Verspätetes Binding für Animation:

- PropertyChanged wird erst später ausgelöst mit Timer.
- Mit 2 Properties LastProjectNote und CurrentProjectNote. Welche zu einem gewissem Zeitpunkt sichtbar sind.

VisualStateManager: GoToState für Wechsel von einem zu anderem State.

Dependency Property: Speichert nur noch das Delta (Unterschied zu anderen). Zuerst wird Dependency Property angeschaut (GetValue), danach wird auf Standardelement geschaut. Braucht weniger Speicher, besitzt Binding.

User Stories

Agenda: Für nächsten Sprint User Stories festlegen

1. Prio: Feedback wenn Filter Kriterium gedrückt wurde
2. Prio: Element erscheint in Filter
3. Pro: Element kann hinübergezogen werden.

Weiteres

Meeting 9.12., genauer festlegen

Nach Montag genauer besprechen.

Protokoll 14.11.11

Erstellt am 14.11.2011, Autor: cheidt

Protokoll abgenommen am 28.11.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Protokolle abnehmen](#)

[Präsentation der User Stories \(Sprint 4\)](#)

Protokolle

Agenda: Protokolle abnehmen

i.o.

Präsentation der User Stories (Sprint 4)

Agenda: Präsentation der User Stories des Sprints 4

Abnahme ok, siehe [User Stories](#)

Protokoll 11.11.11

Erstellt am 11.11.2011, Autor: cheidt

Protokoll abgenommen am 14.11.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Christian Moser, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Besprechung der Planung des Sprints 5](#)

[Layout/Externes Design besprechen](#)

[Termine](#)

[Beschlüsse](#)

Planung Sprint 5

Agenda: Besprechung der Planung des Sprints 5 gemäss [User Stories](#)

Toucherkennung bei 3-D Elementen in der Übersicht (evt. Masterstudent der unterstützen kann). Ansonsten mit Transformation arbeiten anstatt 3-D.

Arbeiten entweder mit 3-D Modell. Elemente müssten 3-D aufgebaut werden (Licht, Texturen etc.). Oder mit Transformationen im 2-D Bereich.

Ziel: Benutzer muss wissen wieviel Project Notes es gibt und wo er sich befindet.

Lösungen skizzieren.

Am Anfang einfache Lösung mit Scrollbalken bei Zeit Ausbau mit 3-D. Wechsel von Übersicht in Detailansicht ist wichtiger.

Test in separaten Projekt und dann Integration bei Erfolg.

Externes Design

Agenda: Layout für die Definition der "Definition of Done" der User Stories besprechen.

Alle Texte nach links ausrichten.

Weniger Tiefe bei den Filterkriterien. Runde Ecken sind nicht unbedingt Zühlke spezifisch aber in diesem Fall ok (Unterscheidung der Elemente).

Kriterienliste näher bei Knopf. Kriterien bewegen sich automatisch zu Filter. Falls alle Slots voll sind bewegt sich Kriterium leicht nach links und rutscht danach wieder zurück. Eventuell Hilfetext in Box als Hilfe.

Wie würde Text über Box genannt werden? z.B. doch Filter?

Filterkriterien bleiben immer alle auf gleiche Art sichtbar (keine leeren Felder). Evt.

ausklappbare Lasche mit Projektdaten. Diese klappt sich ein sobald gezoomt wird.

Schrift verkleinern. Projektdaten erhalten Rahmen, damit es eher einer Lasche ähnelt. Rahmen zwischen Projekt Note und Daten. Einklappbar → als weitere Lösung.

Elemente etwas flacher gestalten (Rahmen nicht 5px).
Minus weglassen.

Termine

SP5: 25.11: 15:00 in Rapperswil

SP6: 09.12: 15:00 in Schlieren (Invite an Markus)

Beschlüsse

- Layout fertigstellen und dokumentieren
-

Protokoll 07.11.11

Erstellt am 07.11.2011, Autor: cheidt

Protokoll abgenommen am 14.11.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Layout](#)

[Weiteres](#)

[Beschlüsse](#)

Layout

Agenda: Layout für die Definition der "Definition of Done" der User Stories besprechen.

Titel weglassen, da bei Zühlke die PNs auch nicht angeschrieben sind.

Hinter dem Tisch mit physischem Element arbeiten.

Liste und nachher Navigation mit Pfeilen → Probleme mit der Semantik

Perspective Wall bei genügend Zeit.

Weiteres

Alles NF-Anforderungen/User Stories testbar, in wie weit muss dies in den Anforderungen sichtbar sein?

Angabe wie man die NF-Anforderungen testen will, und so beschreiben, dass man sie testen kann.

NF-Anforderungen die nicht aus den User Stories, ebenfalls auflisten (z.B. Wartbarkeit). Diese auch so benennen.

Test-Coverage Tool Lizenz für Visual Studio.

Beschlüsse

- Die NF-Anforderungen die nicht aus den User Stories ableitbar sind, werden mit entsprechender Erklärung aufgelistet.
 - Layout weiter ausarbeiten
 - Verfügbarkeit der Coverage Tool Lizenz abklären
-

Protokoll 04.11.11

Erstellt am 04.11.2011, Autor: cheidt

Protokoll abgenommen am 25.11.2011 durch Christian Moser

Anwesend:

Christian Moser, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Code Review](#)

[Detail View und Filter entfernen](#)

[Weiteres](#)

Code Review

Arbeiten mit Prism?

Detail View und Filter entfernen

besprochen

Weiteres

Linien bei den verkleinerten PNs in 2 Elemente teilen

Konzept erarbeiten und versenden damit dieses danach abgenommen werden kann.

Schlichtes Design

Hintergrund in grau und primäre Aktionen in blau oder gelb. Sekundäre Aktionen immer weniger farbig.

Darstellung analog PowerPoint Präsentation

Protokoll 31.10.2011

Erstellt am 31.10.2011, Autor: cheidt, dtreichl

Protokoll abgenommen am 07.11.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Christian Moser (per Skype), Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Verschoben auf 4.11.2011: Codereview mit Christian](#)

[Demonstration der Must User Stories](#)

[Besprechung der Planung des Sprints 4](#)

[Prozessmodell](#)
[Weiteres](#)
[Beschlüsse](#)

Verschoben auf 04.11.2011: Code Review mit Christian

Verschoben auf 4.11.2011, 15.00 Uhr, Gebäude 1
beinhaltet Test-Dokumentation

Demonstration der für den Sprint 3 geplanten Must User Stories

entspricht MS: Architekturprototyp Demo, alle Komponenten & Designentscheide begründet (mündlich)

Kein Fenster für Detailansicht sondern später ein Overlay (Besprechung darüber am 4.11.2011).

Zudem zusätzlicher Controller nötig (erstellt neue Fenster).

Services und Preparer zusammen nehmen? Für Deployment könnten die Assemblies zusammengenommen werden => aktuelle Architektur aber auch ok.

Besprechung der Planung des Sprints 4

Planung gemäss [User Stories](#) i.o.

Wieviel Aufwand brauchen die User Stories für das Filter setzen/entfernen vom Filter in der Detail View? Kann hier nicht das Setzen/Entfernen des Filters in der Übersicht übernommen werden? Soll der Filter in der Detailansicht überhaupt entfernt werden können?

Soll beim Filter entfernen in der Detail View zurück zur Übersicht gegangen werden oder nicht?

Umsetzung mit Wechsel zur Gesamtübersicht, danach testen? Besprechung am Freitag dem 4.11.2011

Prozessmodell

Innerhalb von Prozessbeschreibung kann geschrieben werden, dass die Unterkriterien frei angegeben werden können. Daher gibt es eine gewisse Varianz, welche durch Mapping gelöst werden soll.

Abnahme durch den Kunden fehlt im Prozessmodell. Begründung warum keine weitere Informationen als diese von der Project Note angezeigt wird. Da ansonsten eine weitere Abnahme durch den Kunden entstehen würde.

Weiteres

Annahme für Config File für Aggregation der Tags treffen und dieses dann Christian Moser zusenden, damit es validiert werden kann.

So sinnvoll konfigurierbar und anpassbar für die spätere Verwendung sein. Funktionalität steht im Vordergrund. Speicherung als csv, Textfile oder xml. Eher einfach maschinenlesbares Format, xml bevorzugt.

Bei Auswahl des Filterkriteriums kein "ok" oder "abbrechen" Knopf sondern Klick auf Element giltet als ok (ähnlich DropDown).

Microsoft besitzt PivotViewer für Excel und Silverlight, welcher grosse Datenmengen filtern kann. → Inspirationsquelle : Liste von Suchkriterien auf einer Seite, auf anderer Seite sind alle Entitäten. Gruppierung nach Technologie wird zu Balkendiagramm.

Dokumentieren, warum/warum nicht PivotViewer verwendet.

Die "Definition of Done" bei User Story ist wichtig. Wann ist diese fertig? Was ist das Abgabekriterium? Bei funktionalen Anforderungen noch kein Problem.

Z.B. alle Unit Tests laufen durch.

Oder: Zwei Versionen werden 10 Personen gezeigt, der Favorit ist das Endprodukt.

Oder: 3-5 Varianten werden vorgeschlagen, danach wird eine aus diesen ausgewählt.

Alles was nicht direkt vom Kunden definiert werden kann.

An Meetings so wenig Diskussion wie möglich. Wird erreicht durch die Definition von Kriterien.

Verknüpfung von User Stories über mehrere Sprints mittels Epics.

Beschlüsse

- Architektur: Services und Preparer werden nicht zusammen genommen, da der bestehende Ansatz auch richtig ist.
- PivotViewer: anschauen und dokumentieren, warum nicht 1:1 eingesetzt.
- Filter: Wo setzen/entfernen, Wechsel zur Übersicht? Besprechung am 04.11.11
- "Definition of Done" für User Stories werden erstellt.
- MS: Architekturprototyp Demo, alle Komponenten & Designentscheide begründet (mündlich) -> OK

Protokoll 24.10.2011

Erstellt am 24.10.2011, Autor: cheidt

Protokoll abgenommen am 31.10.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Notizen vom 17.10. besprechen](#)

[Protokolle vom 10.10. und 17.10 abnehmen](#)

[MS: Domain Model \(formelles OK Stolze \(UML\), inhaltliches...](#)

[Beschlüsse](#)

Notizen vom 17.10. besprechen

- Umfangreiche Dokumente im vornherein schicken, zur Vorbereitung.

- z.B. Begriff Behavioural Variables nicht bekannt für Aussenstehende, beim Vorbereiten überlegen ob solche Begriffe bekannt sind. Erklären oder weglassen.
- Bei "Kundenmeeting" die User Stories anders vorbereitet (in Form von z.B. Karten, die danach sortiert werden). Mit weniger Detailsangaben. Führt zu weniger Ablenkung, können noch einmal erklärt werden. Bei digitalem Dokument sind nicht alle auf einmal sichtbar.
 - Aktuelle Liste schicken an Kunden, damit sie sich die Liste im Vorherein ansehen können
 - Mit Christian besprechen wegen User Stories Liste
- Übertitel klarer definieren (welcher MS wird besprochen?)
- Begrenzten Zeitrahmen aufstellen für zu besprechende Themen (bei Prototyp). Bei Protokoll gleich mitaufschreiben.
- Titel der User Stories ist zu sehr auf Implementation fixiert anstatt auf Bedürfnis

Protokolle vom 10.10. und 17.10 abnehmen

i.o.

MS: Domain Model (formelles OK Stolze (UML), inhaltliches OK Auftraggeber)

Filepath bezieht sich etwas zu stark auf Implementation. Der Kontext ist mit diesem Domain Model nicht ersichtlich.

Es fehlt der Ersteller der Project Note. Durch welchen Prozess läuft so eine Project Note? Wie läuft Erstellungsprozess (wer ist Autor? Holt er Input bei Kollegen? Wird es abgesegnet? Wird es gedruckt? In welcher Form liegt das Dokument vor? Wie kommt dies in das Format, welches gewünscht ist? Woher kommen die Bilder? Welche Schritte bis der Kunde die PN in der Hand hält. Was muss Sekretariat alles machen? Als Analysebeschreibung, Prozessmodell).

Diese Informationen zusätzlich zum Domain Model.

Domainmodel ist nahe an einem Glossar: Was ist das Wichtigste in einer Domäne?

Darstellung des Domain Models mit dem Visual Studio gilt als valides UML (wäre erst unkorrekt, wenn die Pfeile in falsche Richtung gehen würden etc.)

Neuer Untertitel bei Domain Analyse (z.B. aktueller Ablauf).

Beschlüsse

MS: Domain Model (formelles OK Stolze (UML), inhaltliches OK Auftraggeber) -> OK von Markus Stolze (24.10.11), OK von Christian Moser per Mail (25.10.2011)

Protokoll 17.10.2011

Erstellt am 17.10.2011, Autor: Ielmer, cheidt

Protokoll abgenommen am 24.10.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Marco Balzarini, Christian Moser, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Vision und Personas](#)

[Papierprototyp & Navigationkonzept](#)

[User Stories](#)

[Risikomanagement](#)

[Kommende Sitzungstermine](#)

[Beschlüsse](#)

Vision und Personas

MS: Vision Dokument V1 (mit Personas und Key Szenarios) (formelles OK Stolze, inhaltliches OK Auftraggeber)

Vorzunehmende Anpassungen:

- Arbeitskontext bei Bettina Bewerberin anpassen (nicht als Arbeitsumgebung betiteln).
- Technische Expertise ersetzen durch Dinge, die sie in der Hochschule gelernt hat. Was kennt sie für Tools, mit was kann sie programmieren? Eher Techie oder Anwender?
- Behaviour Variables → Verhaltensvariablen
- Sie stöbert durch die verschiedenen Project Notes → beschreiben wie dies effektiv umgesetzt wird, solange die PNs in Papierform vorliegen.

Was könnte für Bewerber interessant sein, um bei PNs zu suchen. z.B. Rollen oder welche Teile hat Zühlke gemacht. Was für Projekte könnten auf einem zukommen. Ist vorallem interessiert an Software Solutions (Services).

Ideale Umsetzung von Ideen, auch wenn die danach nicht verfügbar sind.

Anderer Einstieg für den Bewerber, welcher aber dann in der Umsetzung ignoriert wird. Zwei Suchvarianten anbieten?

Gewichtung zwischen den verschiedenen Personas. Vorallem bei widersprüchlichen Interessen. Jeder Besucher bekommt Besucherbadge, welcher je nach Besuchertyp anders getagged ist.

Mit getaggten Objekt können verschiedene Informationen aufgerufen, welche sich auf Personas fokussieren.

Bewerber Badge → Infos über Software Solutions

Ohne getaggte Objekte müssen Einstiegsstiegepunkte selbst gewählt werden.

Papierprototyp & Navigationkonzept

Kurze Vorstellung der Ergebnisse des Papierprototyps und des Navigationkonzeptes.

Beobachtung beim Test:

Fragt Person an, die an diesem Projekt gearbeitet hat. Will wissen, ob es wohl noch mehr Swisscom Projekte gibt. Sucht Swisscom über den Filter und bekommt neue Anzeige. Setzt den .Net Filter auch direkt oben und kommt so zu seiner Information.

Informiert sich über die weiteren Informationen die angehängt sind. Findet diese überflüssig.

Druckfunktionalität/Verschickenfunktionalität fehlt ihm.

Löscht beide Filter.

Möchte eine Point of Sale Applikation anschauen.

Ideen/Input:

Auszoomen ist eher untypisch. Die Übersicht ist immernoch zu 80% sichtbar, damit klarer ist, dass sich noch etwas hinter der Detailansicht befindet (Mantra for Information Searching).

Beim Schliessen geht das Element auch wieder zurück. Project Note in Mitte nehmen?

Bewegung in alle Richtungen.

Grosses Drop Down anstatt Kontextmenu.

Das erste, was man zeigen will, ist die Menge. Falls angezeigte Project Note mir nicht gefällt, verliere ich mein Interesse.

Anderer Demo-Modus? Nicht so hohe Priorität (klassischer Bildschirmschoner?).

Kategorien schon auflisten links von Raster. Attraktivität des Filters steigern. Was erwartet den Benutzer? Hemmschwelle senken.

Text für Filter anpassen. Titel anpassen?

Titel und Filter trennen, damit Leute wissen, was dieser genau macht.

Nicht alle Themen sind relevant als Filter. Themen zusammennehmen? Wie wird dies erwartet? Lookuptabelle die selber merged. Neu hinzugekommene Technologien werden einzeln oder gar nicht aufgelistet.

Die meisten Benutzer werden mehr browsen und nicht spezifisch suchen. Automatisches Bearbeiten der Metadaten, Exoten dürfen auch wegfallen. Einzelne Überkategorien dürfen auch wegfallen. Kein Nachführen der Metadaten. Mut zur Lücke. Gesamte Anzahl auch reduzieren.

Config-File wäre wünschenswert.

Freitext Suche (Erweiterung)

Anzahl Filter begrenzen

User Stories

User Stories durchgeschaut und priorisiert.

MS: Iteration 2 Feature Set (oder User stories) festgelegt (alle Features; Meeting mit Auftraggeber; Review mit Stolze),

MS: Super-Set der "User Stories" (vollständiges Backlog, Sammlung von möglichen Anwendungs-Features mit Klassifizierung als Must/Should/Could) entwickelt (formelles OK Stolze, inhaltliches OK Auftraggeber),

siehe:

https://docs.google.com/spreadsheet/ccc?key=0Aubz_al1gBrdHI2MUtmVHhpMmRaZTVuUXVMbFhUZ3c&hl=en_US#gid=0

Tags: Über ähnliche navigieren und Zusatzinformation

Lieber eine Navigationsmöglichkeit umsetzen mit grafischer Ausarbeitung, anstatt beide umsetzen und dann beide nur halbfertig.

Kontextmenu bei Tag nötig? Oder auf andere Weise umsetzen? Das welches am ehesten erwartet wird?

→ Am Anfang immer alle Filter löschen (vereinfachte Methode).

Neue User Story: Volltextsuche (Titel und Technologien zusammenhängen) → billiger Filter

Filter highlighthen.

Eselsohr bedeutet Rückseite. Andere Animation für Wechsel.

Bedürfnis bei User Stories nicht Lösung. Dann muss Eselohr Variante nicht zu weit angepasst werden.

Immer anzeigen, wie viele Notes von Maximum angezeigt werden (Bsp. 20 von 375 Projekte)

Eine User Story für Implementierung & Gestaltung? → Anzeigen, welche User Stories eigentlich zusammen gehören (aber keine Unterteilung)..

Bringt Reaktion auf nicht klickbare Elemente etwas? Ausprobieren und daher umsetzen.

Position aus PDF ausschneiden und in Bild umwandeln. Fixe Zeit vornehmen um dies zu testen. Wenn dies nicht möglich ist, entscheiden, was man sonst tun soll. Einen Tag dafür einplanen.

Aufteilen in 2 User Stories. Einmal auslesen und einmal darstellen. In Sprint 4 umsetzen.

→ Wird in SP3 umgesetzt, wegen erhöhtem Risiko.

Badgerkennung bei User Stories hinzufügen.

Verschicken und Drucken gut als Erweiterung für später.

Risikomanagement

Neue Risiken vorstellen.

Kommende Sitzungstermine

Termine definieren für die nächsten 6 Wochen (neben den wöchentlichen Meetings mit Markus).

Vorschlag:

31.10.: SP3, Codereview mit Christian, Demonstration der Must User Stories, HSR, 14:00-17:00

Markus Stolze von 15:00-16:00, dafür Termin von 13:00-14:00 auslassen

11.11.: SP4, Demonstration der Must User Stories, HSR, 13:00-15:00

Markus Stolze von 13:00-14:00

21.11: Falls nötig Termin definitiv vereinbaren (Skype)

TBD:

28.11.: SP5, Demonstration Prototyp, evt. Repriorisierung oder alternative User Stories, Zühlke Schlieren, 15:00-17:00

Beschlüsse

- MS: Vision Dokument V1 (mit Personas und Key Szenarios) (formelles OK Stolze, inhaltliches OK Auftraggeber)
-> OK

- MS: Iteration 2 Feature Set (oder User stories) festgelegt (alle Features; Meeting mit Auftraggeber; Review mit Stolze)
-> OK, Anpassungen noch vornehmen
 - MS: Super-Set der "User Stories" (vollständiges Backlog, Sammlung von möglichen Anwendungs-Features mit Klassifizierung als Must/Should/Could) entwickelt (formelles OK Stolze, inhaltliches OK Auftraggeber)
-> OK, kann bei Besprechung des Sprints 3 (wieder) ändern
-

Protokoll 10.10.2011

Erstellt am 10.10.2011, Autor: cheidt

Protokoll abgenommen am 24.10.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Besprechung des Creative Workshops](#)

[Planung letzter Sprint](#)

Creative Workshop

Bei Demo-Modus Stichwörter anzeigen. Weitenwirkung? Bilder/Ausschnitte auf Project Notes vergrößern? Überlegung zu ausgewählten Navigationsart.

Testen an Touch Bildschirm: Wie liest man die Project Note, sieht man das Ganze? Kann gescrollt werden oder ist Zoom möglich. Tisch eventuell anheben an einer Kante (30-45 °).

Anordnung mit:

- Multidimensional scaling
- 3-D Hyperbolic Tree
- <http://well-formed.eigenfactor.org/index.html>
- Cone tree

Planung letzter Sprint

Poster und Präsentation sind extra Stunden und können später abgegeben werden daher nicht im letzten Sprint mitgezählt.

Protokoll 03.10.2011

Erstellt am 03.10.2011, Autor: cheidt

Protokoll abgenommen am 10.10.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Lukas Elmer, Christina Heidt, Delia Treichler, (Christian Moser, Marco Balzarini)

Agenda:

[Fragen Scrum](#)

MS: Architekturprototyp Demo, alle Komponenten & Designentscheide begründet (mündlich)

MS: Vision Dokument V0 verabschiedet mit Preliminary Personas & Szenarios

MS: Projektplan inkl. Plan für Benutzerbeobachtung (OK des Kunden) und Aufgabenteilung (mindestens 4 Milestones), Sinnvolle Arbeitspakete pro Milestone mit soll und ist Aufwänden

MS: Benutzerbeobachtung/Befragung beendet

MS: Usability Tests

Weiteres

Beschlüsse

Fragen Scrum

Keine Fragen im Moment

MS: Architekturprototyp (Demo)

Architekturprototyp prinzipiell noch zu früh.

Automatisches PDF einlesen als Anforderung. Noch weitere Prototypen um andere Probleme zu lösen. Noch kein Architekturprototyp im eigentlichen Sinn. Nur falls es wirklich nur um PDF anzeigen geht. Sobald klar ist, was es alles können muss, kann ein Prototyp mit den wichtigsten Funktionen erstellt werden (von reinladen bis zu dem Punkt, dass der User es verwendet). Evt. wird noch weiteres Framework oder ähnliches benutzt. Diese müssen im endgültigen Architekturprototypen genutzt werden (welche danach im Architekturdokument festgehalten werden). Noch nicht als MS abgesegnet, da noch etwas früh (ca. 2-3 Wochen).

Anmerkung Christian & Marco: Je nach dem wie Navigation funktioniert, muss die Architektur dementsprechend angepasst werden.

MS: Vision Dokument V0

Im Moment abgeschlossen. Vielleicht kommen im Verlauf des Projekts noch weitere verfeinerte Ideen zu Szenarien.

Weitere Möglichkeit zum Validieren: Auftraggebern die Szenarien vorlegen und befragen ob dies soweit Sinn ergibt.

- Vision: Erweiterungen aus der Aufgabenstellung. Auf was wird die Software implementiert. Informationen über die nichtfunktionalen Anforderungen (z.B. sollte einfach zu bedienen sein, auch für Leute, die noch nie mit einer solchen Hardware gearbeitet haben). Schon einmal Vorgriff auf Personas.
Anmerkung Christian & Marco: Vision passt. Kunden sollen einerseits den Zugang haben und andererseits will sich die Zühlke auch als innovativ zeigen.
- Project Flip 1.0: Wichtigste Erfahrungen wurden aufgelistet
- Persona 1: Ulrich Umsetzer: Funktion und Arbeitsstil, Pain Points weglassen (da nicht vorhanden für dieses Projekt). Ansonsten i.o.
Anmerkung Christian & Marco: Foto bei Personas hinzufügen um sich diese besser vorstellen zu können.
 - Ist-Szenario-1: Könnte Project Notes mit seinem Smartphone abfotografieren?

- Soll-Szenario-1: Email Adresse eingeben? Businesskarte oder ähnliches abfotografieren? Würde aber danach über eine Nachbearbeitung laufen. Von Applikation an Empfang senden und Empfang verarbeitet diese Information.
→ Definition von User Story.
Anmerkung Christian & Marco: Problem bei Email Adresse angeben. Die Eingabe dauert zu lange, wenn Person abgeholt wird.
Andere Möglichkeit: Dame vom Empfang sucht in der Zeit die Project Notes heraus, welche von dem Mitarbeiter ausgeschaut wurden. Personen müssen sich ja auf Liste eintragen. Dame kann Person eine getaggttes Objekt mitgeben mit welchem er identifiziert werden (Problem, dass sich erst alle anmelden wenn alle Personen da sind). Kombinierte Dienstleistungen. Physischer Akt hinzufügen (Neugier wecken).
- Persona 2: Erik Entscheider: i.o. Ausdrucksmöglichkeit gut als Idee. Für Gesprächsweiterführung.
- Persona 3: Leute, die sich Zühlke als Arbeitgeber vorstellen könnten (Bewerber) und daher auch an der Firma interessiert sind. Diese kommen (hoffentlich) genug früh. Finden von spannenden Themen für das Bewerbungsgespräch. Sicher auflisten in Doku.

MS: Projektplan

Für Sprintende muss Akzeptanzrahmen aufgestellt werden. Dieser muss mit der Zühlke oder Markus Stolze validiert werden. Mit den Milestones und Tickets wird ein Sprint dementsprechend abgenommen. Pro Sprint die Akzeptanzkriterien definieren (User Story kommt in mehreren Sprints vor mit verschiedenen Kriterien).

Benennung: anstatt "User Stories priorisieren" - "User Stories priorisiert", "Risikomanagement ist definiert/nachgeführt" etc.

Letzter Sprint sollte klarer sein, wegen der Dokumenterstellung (von hinten auffüllen). Ab wann arbeiten wir nicht mehr inhaltlich sondern definieren nur noch Dokumente? Was wird am Ende alles gebraucht?

MS: Benutzerbeobachtung/Befragung beendet

i.o.

MS: Usability Tests

Plan für Benutzerbeobachtung → Usability Test. Schnelle Überlegung auf welche Art dies zu lösen ist. 2 Szenarien:

- mit Surface Hardware
- ohne Hardware, sondern mit Dell Touch Screen.

Plan → Wann wir Usability Tests durchführen wollen? Als abschliessenden Test oder ob Schlüsse daraus gezogen werden können sollten. Paperprototype gehört auch zu Benutzerbeobachtung.

Weiteres

Bei Sitzung können auch weniger Teammitglieder vorbeikommen.

Dokumente mitschicken ist nicht nötig, da Markus Stolze wohl selten kurzfristig Zeit hat um diese zu lesen. Bei grösseren Dokumenten bei denen eine Vorbereitung dringend notwendig ist, sollten diese mindestens 2-3 Tage vorher verschickt werden.

Anmerkung Christian & Marco: Anbindung an Sharepoint bringt eventuell Probleme. Dies mit in die Liste aufnehmen und Problem in Risikomanagement eintragen.

Beschlüsse

- Architekturprototyp zu späterem Zeitpunkt vorführen
- Persona 3 hinzufügen
- Personas: Funktion und Arbeitsstil, Pain Points weglassen
- Szenarien um verschiedene Endvarianten erweitern (Problem mit der Email Adresse anders lösen)
- Pro Sprint werden die Akzeptanzkriterien definieren (User Story kommt in mehreren Sprints vor mit verschiedenen Kriterien).
- Ticket Benennung anpassen ("User Stories priorisieren" → "User Stories priorisiert")
- Sprints von Ende her auffüllen mit Dokumentationsarbeiten
- Plan für Usability Tests (mit oder ohne Hardware)
- Wichtige (lange) Dokumente 2-3 Tage früher an Markus Stolze versenden

Protokoll 26.09.2011

Erstellt am 26.09.2011, Autor: cheidt

Protokoll abgenommen am 03.10.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Bewertungskriterien](#)

[MS](#)

[Projektplanung](#)

[Agenda](#)

[Personas](#)

[Weiteres](#)

[Beschlüsse](#)

Bewertungskriterien

Ähnlich wie SE2, bis Architekturprototyp rumtesten und danach Architektur fix definieren.

Gewichtungsfragen werden später besprochen

Dokumente nach SE Dokumentation benennen -> Einfachheit

Bearbeitete Bewertungskriterien-Liste an Markus Stolze schicken

Allg. Fragen

- Was bedeutet Milestone (MS): In einem Protokoll muss stehen, dass dieses Thema in einer Sitzung abgehandelt wurden. Jeder Kriterienpunkt, welcher mit MS gekennzeichnet ist, muss an einer Sitzung besprochen werden.
- Mit möglichst wenigen Iterationen abgenommen: Alles ist so schnell wie möglich ok (beim ersten Mal der Abgabe), falls nicht gibt es Abzug. Keine Dokumente vergessen.

1. Organisation & Durchführung (MS = Milestones; Com=E-Mail)

- Punkt 3: Solange keine Surface 2 Hardware vorhanden ist bis Ende der Arbeit, wird dieser Punkt auf 0 gesetzt.
- Punkt 4: Da gewisse Vorplanungen vorhanden sind ähnlich wie bei RUP. Wenn Mix, dann muss dies in einem Kapitel beschrieben werden (diese Elemente wurden von RUP genommen diese von SCRUM).
- Punkt 5: Redmine reicht. Am Montag dem 03.10.2011 den Projektplan besprechen. Projektplan braucht mindestens 4 Iterationen.
- Punkt 6: Vision etwa eine halbe Seite bis ganze Seite
- Punkt 8: Wenig Personas und wenig Szenarios. Eher Benutzungssituationen anstatt Szenarios (z.B. Zühlke Mitarbeiter kommt dazu und erklärt). Personas am Montag dem 03.10.2011 anschauen. Personen primär charakterisieren und These validieren.
- Punkt 9: Inhaltliches Ok von Arbeitgeber: An Zühlke ein Mail schreiben oder Treffen vereinbaren und dies validieren lassen. Was ist bei User Stories wichtig was weniger wichtig? Markus zeigen was dabei herausgekommen ist. Wird in Protokoll festgehalten. Erklären wieso was gewählt wurde (aus Interviews). Priorisierung vorschlagen.
- Punkt 10: Reduzierte Gewichtung von Domain Model ist ok. Domain Model ist eher als Glossar zu verstehen (mit was wird gearbeitet, z.B. ProjectNotes und dann gleich ein Foto dazu). Weitere Elemente wie Berater oder Technologie (Meta Daten). Um sich Vorstellung zu machen, was alles gebraucht wird, um was gehts in der Domäne? Am Besten als UML als Übersicht und danach eine Beschreibung mit Bild und Text. Soll Leserorientiert sein (Anfang vom technischen Bericht).
- Punkt 11: Feature Set gehört in die Iteration 2.
- Punkt 13: Tickets im Redmine ok
- Punkt 14: Mit Markus Stolze mündlich in einem Meeting besprechen. Protokoll nicht nötig
- Punkt 15: Inhaltsverzeichnis vorstellen, was haben wir konkret geplant, was wollen wir machen in welcher Reihenfolge. Einzelne Dokumente können separiert werden, müssen am Schluss als PDF abgegeben werden.
- Punkt 17: Reicht in Redmine und für Bericht Piechart exportieren und sagen wo man sich verschätzt hat.
- Punkt 18: Change Message: ignorieren. Redmine-Wiki reicht, einfach aktuelles Protokoll noch in Email kopieren. Redmine hin und wieder anschauen bei verschiedenen Task (keine Abgabe von wöchentlichen Dumps).
- Punkt 23: Datum des ok: immer das der darauffolgenden Sitzung
- Punkt 24: Code und Dokumente Reviews. Code-Reviews auch in Code protokollieren (Datum).

- Punkt 26: Tools nur auflisten, nicht weiter beschreiben
- Punkt 28: Gewichtung auch 5. Konkrete Gewichtung gegen Ende der Arbeiten

2.1 Generelle Kriterien für den Bericht

- Punkt 1: HTML Inhaltsverzeichnis nicht nötig. Navigierbares Inhaltsverzeichnis -> gute beschriftete Ordner
- Punkt 2: Lizenzvereinbarung wird benötigt. Damit Zühlke weiss was sie besitzen. Ist alles was wir machen open source? Wie wird dies geregelt? Nutzungsrecht auch für Unterricht, weitere Projekte dürfen darauf aufbauen. Mit Marco und Christian besprechen.
- Punkt 13: ACM = American Society Computer Machinery (www.acm.org). Library über Papers. Punkt wird verlangt. Paper raussuchen zu unserem Thema

2.2 Kriterien Abstract / Management Summary / Extended Management Summary Struktur

Verschiedene Zielgruppen bei Abstract, Management Summaries (jemand von aussen der sich detailliert informieren will mit Fokus auf Business Value), Technischer Report (für Leute mit technischem Hintergrund), Projektdokumente für Maintenance.

Management Summary: Nicht mehr Text als in Abstract. Die Vision kann auf Management Summary verweisen.

Extended Management Summary: In Dokument ist länger als Management Summary. Damit man gut versteht was gemacht wurde. Etwa 2-4 Seiten

2.3 Qualität weiterer Deliverables (auch Teil des Bereichs "Bericht")

- Punkt 1: Problemstellung lässt sich ja darstellen (mit Bild von Surface 2). Demonstration auf Bildschirm wie man gewisse Funktionen lösen kann (Minidemo einfach auf normalen Bildschirm falls Hardware nicht verfügbar).
- Punkt 2: Extended Management Summary benutzen um damit ein externes Wiki zu führen (unter wiki.hsr.ch). Dort für Studienarbeit diese erstellen und am Ende erstellen. Die Wiki Seite muss angeschaut werden.
- Punkt 3: Brochure Text = Management Summary
- Punkt 5: Poster kann auch nach Abgabe erstellt werden von allem anderen. Um zu üben wie Poster gemacht werden. Evt. mit Präsentation der Arbeit inkl. Feedback.

3.0 Kriterien Lösung Generell (Bereich "Inhalt")

Punkt 2: i.o.

3.1.1 Kriterien Vision Dokument / "Projektantrag" (SE Dok: "Vorstudie")

- Punkt 2: Benutzerproblem kann verschieden definiert werden. Wer ist der Benutzer? Ist Person am Tisch der Benutzer? Wenn ja ist das Benutzerproblem wie er die Zeit gut nutzen kann. In Vision dokument ist-zustand beschreiben, Zühlke will Wartezeit sinnvoll nutzen. Ist im Interesse des Besuchers sich über die Zühlke zu informieren. Bekommt Gefühl was Zühlke alles macht.
- Punkt 3: Stakeholderanalyse: Wer braucht es und wer sind die indirekten Nutzer

- Punkt 4: Konkurrenz ist bestehende Lösung (Papier).

3.1.2 Personas & Szenarios & User Analyse Dokument (SE Dok: "Vorstudie")

- Punkt 2: Behavioural Variables überlegen und diese extrem legen (keine Durchschnitts-User). Jemand mit speziellen Eigenschaften
- Punkt 5: Nicht optional ist i.o.
- Nachtrag fokussierte Personas:
 - Seid mutig bei der Auswahl der Nutzergruppe:
 - Sagt zum Beispiel: wir fokussieren auf Techno-Affine Nutzer mit Smart-Phone Erfahrung
 - Klar, nicht alle Zühlke Besucher passen auf das Profil, aber die Anwendung lässt sich viel besser FOKUSSIEREN / designen.
 - Vielleicht fallen euch andere/weitere/bessere Einschränkungen ein (Die Kunst ist natürlich Zühlke dann davon zu überzeugen, dann wenn man fragt wollen Auftraggeber immer eine Anwendung "für alle" (ohne zu realisieren, dass gleichbedeutend ist mit langweilig und nur mässig nützlich und nur mässig bedienbar)

3.1.3 Kriterien Anforderungen / Software Requirements Dok (SE Dok: "Anforderungen")

- Punkt 2: Dokument für Anforderungsspezifikation mit Liste der User Stories
- Punkt 5/6: Gewichtung auf 0
- Punkt 7: Design Constraints: Es wird mit Surface 2 Tisch gearbeitet oder PDF File als Vorlage. Alle Architekturentscheide, die wir nicht mitbestimmen können.

3.1.5 Kriterien Domain Analyse Dok (SE Dok: Domain Analyse: GUI)

- Punkt 3: Falls es mehrere Screen gibt braucht es ein Navigation Map
- Punkt 5: Wie kann Konzept möglichst einfach genutzt werden? Muss nicht zwingend ein Papier Prototyp sein. Screens mit Photoshop malen und durchtesten mit Beamer.
-> Denkfehler möglichst früh eliminieren.
- Punkt 6: Dokumentieren, dass Teammitglieder die Guidelines gelesen haben. Danach schauen welche nicht eingehalten wurden und wie schlimm ist dies.
- Punkt 7: Farbenblindheit (Check mit Farben). Reicht bei Accessibility.
- Punkt 8: Abnahme-Sitzung zu jedem Sprint inkl. Abnahmedokument (z.B. was verschoben wurde und wieso).

3.2.2 Kriterien Architektur & Dok (SE Dok: "Entwurf")

- Punkt 2: Design Entscheide müssen dokumentiert sein. Alles was gemacht wurde muss Sinn machen/begründet sein
- Punkt 3: NDepend vielleicht nötig. Package Dependency Übersicht ist sehr wichtig. Dependency berechnen mit NDepend (Plugin evt. von Recharper).
- Punkt 5: Visual Studio reicht
- Punkt 6/7: Arch Decision: Architekturentscheidungen

- Punkt 8: Sensitivitätsanalyse: Bei veränderter Gewichtung starke Änderung des Ganzen, dann ist es sehr sensitiv. Plausibel machen, dass man es auch auf andere Weise lösen könnte. z.B. wegen Dokumentdarstellung
- Punkt 10: Kann auch mit 0 belegt werden. Angemessenheit der Architektur beschreiben. Funktionen und gewisses Modul dafür zuständig.

3.3.1 Kriterien Test & Dok (SE Dok: "Realisierung & Test")

- Punkt 1: Anschnitt: Testbarkeit des GUIs. Schreiben was man mit Unit Tests testen will.
- Punkt 4: NF Anforderungen: was nicht getestet wird wird auch nicht aufgelistet.
- Punkt 5: Punkt löschen

3.3.2 Kriterien Code & Doc (Bereich "Realisierung & Test")

- Punkt 1: Sofern ein Checkout ohne Probleme funktioniert dann gut (beschreiben was es alles braucht an Software).
- Punkt 2: Markus Stolze hat auch einen Windows. Tests durch Screenshots dokumentieren.
- Punkt 4: Wegen Doc mit Michael Gfeller reden.
- Punkt 6: Resultate der Metrik Auswertung.
- Punkt 7: Screenshot welcher in der Dokumentation sein muss
- Punkt 8: Irgendwo muss geschrieben werden, dass ein gewisses Profil vom Resharper verwendet wird (ink. Screenshot, dass man sich daran hält).

3.3.3 Kriterien Benutzer & Installationsdokumentation (Bereich "Realisierung & Test")

Installationsanleitung für Programm (sofern Hardware bis dann vorhanden).

Installationsanleitung für Importieren des Projekts. Benutzeranleitung darf nicht nötig sein.

MS

Milestone MS: SE Model (UP; Scrum; ...) festgelegt und entsprechend durchgeführt besprochen

Beschluss: Konzentration auf Scrum, einige Elemente aus RUP. Genauere Erläuterung im Dokument Projektmanagement, Kapitel Vorgehensmodell

Projektplanung

Projektplan von einzelnen Sprints bestehen lassen (Screenshot oder Ähnliches).

Solange sich nur Features und keine Projektdaten verschieben ist Projektplan in RedMine ok.

Sobald Zeiten verschoben werden muss ein Screenshot gemacht werden.

Am Ende am Besten Übersicht wie sich Tickets verschoben haben -> Abklärung von Lukas Elmer wie dies mit Redmine gemacht werden kann.

Pro Sprint wichtigste User Stories definieren -> in einem Sprint implementieren können

Crashkurs in SCRUM am Montag dem 03.10.2011. Buch/Tutorial anschauen und danach Fragen stellen

Agenda

Agenda verschicken

1. Punkt: Minutes vom letzten Mal durchschauen und besprechen, erinnern an was wir gearbeitet haben, gegebenenfalls besprechen.

Link zum letzten Protokoll verschicken.

Protokoll alles auf einer Seite (ein Dokument), aktuellstes Protokoll zuoberst.

Personas

Arbeitskontext: Personen die sich für Themen interessieren, Arbeitsstil (stärker Personen-Relationen interessiert. Wo hat mein Berater gearbeitet. Personen oder fachlich interessiert).

Interaktionsstil: Eher visuell oder sprachlich orientiert -> Behavioural Variables. Diese sind wichtiger als beispielsweise Arbeitskontext.

Weiteres

In Projektreflection schreiben wo man sich am meisten geschätzt hat.

Beschlüsse

Vorgehensmodell: Konzentration auf Scrum, einige Elemente aus RUP.

Projektplan: Bei Verschiebungen von Sprints (z.B. Verlängerung) Screenshot als Dokumentation

Protokoll 19.09.2011

Erstellt am 19.09.2011, Autor: dtreichl

Protokoll abgenommen am 26.09.2011 durch Markus Stolze

Anwesend:

Markus Stolze, Christian Moser, Michael Gfeller, Lukas Elmer, Christina Heidt, Delia Treichler

Agenda:

[Dokumente, Deliverables](#)

[Termine](#)

[Vorgehen \(Scrum, RUP\)](#)

[Technologie](#)

[Resharper](#)

[Tools](#)

[Beschlüsse](#)

Dokumente, Deliverables

Dokumente sind ein Mix zwischen den Anforderungen von Zühlke und der HSR.

Dokumentation

- Ideen: Protokollieren, dass nicht geht. Das bedeutet, das wir daran gedacht haben, es aber nicht geht.
- Wo hat es funktioniert, wo nicht.
- Video: 2 min, Problem beschreiben, soll veröffentlicht werden sollen dürfen

Projektplan, Vorgehen: Dokumente nachführen. In jedem Sprint die Nachführungen am Projektplan durchführen.

Zur Durchführung

- Funktionen und Darstellung von Flip 1.0?
Dokumentieren, damit Fehler nicht nochmals geschehen. Ev. können diese Ideen uns weiterhelfen.
- Ideale Lösung vs realistische Lösung. Gründe angeben, warum nicht die ideale Lösung umgesetzt. (Zeit, Architekturentscheide etc..)
- Wichtige Informationen erhalten wir von indirekten Benutzern (was sind Project Notes, was ist der Inhalt)
Interview Termine mit Marketing, Christan an einem Nachmittag. Nebenbei Ausschau halten für Kunden.
Was macht ein Kunde jetzt vor den Project Notes?
Am besten Kunde, den Zühlke kennt, den wir 10 min fragen könnten. Kunden fragen, welche Project Notes sie in letzter Zeit angeschaut haben. Videokamera installieren und schauen, was Kunden machen.

wenn keine reale Beobachtung möglich: Zühlke befragt Kunden, ob nach Project Notes umgesehen.

Ziel:

Project Notes darstellen und zeigen, was mit diesen Geräten und der Technologie gemacht werden kann. Für Kunden interessant.

Termine

Für Fragen zwischendurch können wir gerne auch ein Email schicken. Technische Fragen: Michael, Christian.

Administrative Statusmeetings im 1-2 Wochen Rythmus.

- Abgabe 23.12.11, 12 Uhr

- Wöchentlich 1 h mit Herrn Stolze am Mo, 13.10-13.55, Ausnahme: 26.09.11: Meeting um 10:10-10:55
- Termin vom 30.9 auf 3.10. An diesem Tag sind Christian und Marco ab 16.00 nicht mehr verfügbar. Wir müssen die Zeit noch wählen und mitteilen.
- Termin vom 21.10 auf 24.10 verschoben. 13.00-16.00

Weitere Termine (ab Ende Oktober) sind noch nicht fest.

Vorgehen (Scrum, RUP)

Kriterienliste: Anpassen, was was bedeutet nächstes Mal fragen

Technologie

Surface2 oder 1?

Grosser Unterschied ist die Auflösung. Surface 2 hat FullHD.

Programm für Surface 2 kann bei Nicht-Eintreffen des Tisches auf HP Touchbildschirm angeschaut werden. Aber nur 2 Touchpunkte. Grösse und Auflösung ähnlich. Ev. auf Tablet testen.

-> Entscheid fällt auf Surface2, da es genügend Möglichkeiten gibt, das Programm zu testen.

Richtlinien sagen, dass von jeder Seite bedienbar, auch mehrere Leute miteinander.

Resharper

@Herr Stolze: Können Sie uns diese Frage beantworten? Oder müssen wir hier Zühlke fragen? Könnten Sie ggf. eine Classroom Licence beantragen?

<http://www.jetbrains.com/eforms/classroomRequest.action?licenseRequest=RSCL>

Was ist das Resultat der Arbeit? Prototyp oder bereit zur Ausstellung.

Im Bezug auf Lizenzen als Studienarbeit anschauen -> nicht kommerzielle Resharperlizenz.

-> Michael organisiert sie für uns (Classroom-Lizenz).

Demoversion für 30 Tage zur Überbrückung

Versionierungstool, Projektmanagementtool

- Wir nutzen Redmine und SVN. Zudem ist die Versionierung im Redmine gut integriert.
- TFS hat bessere Integration in VS, BuildServer ist incl. Michael könnte uns den TFS für die Versionsverwaltungs zu Verfügung stellen.

-> Der Entscheid fällt auf Redmine und SVN.

Lukas legt Michael einen Redmine-Account an.

Integration Subversion in VS: Visual SVN (kostet), Tortoise SVN für VS

Benötigen wir einen Buildserver oder ist der Overhead zu gross? ->Nein, wir benötigen keinen Build-Server

Beschlüsse

- Technologie-Entscheid fällt auf Surface2
- Classroom-Lizenz für Resharper wird beantragt
- Redmine und SVN als Projektmanagement- und Versionierungstool

Änderungsgeschichte

Datum	Version	Änderung	Autor
23.09.2011	1.0	Erste Version des Dokuments	dtreichl
25.09.2011	1.1	R4 hinzugefügt + Werte angepasst	lemer
25.09.2011	1.2	R4 bereinigt	lelmer
30.09.2011	1.3	Review	cheidt
03.10.2011	1.4	Neues Risiko R5 gemäss Skype-Meeting	lelmer
16.10.2001	1.5	Neues Risiko R6	dtreichl
17.10.2011	1.6	R1, R6 sind eingetreten und somit bereinigt, R2 ist mit den User Stories wahrscheinlicher geworden, Layout angepasst.	lelmer
23.10.2011	1.7	Neues Risiko R7, R2 Wahrscheinlichkeit von 20% auf 15% angepasst.	lelmer
31.10.2011	1.8	R7 bereinigt	lelmer
14.11.2011	1.9	Schadenspotenzial R2 angepasst, R3 bereinigt	lelmer
28.11.2011	1.10	keine Nachführung nötig gewesen	dtreichl
13.12.2011	1.11	R2 bereinigt, keine Risiken mehr vorhanden	lelmer

Risikomanagement

Nr.	Risiko	Beschreibung	W'keit des Eintretens [%]	Schadens-potenzial [h]	Reserven [h]	Vermeidungs- und Verminderungs-massnahmen	Aktionen beim Eintreffen	Sprint neu	Sprint bereinigt
1	Surface2 wird nicht rechtzeitig geliefert	Der Touchtisch kann nicht rechtzeitig geliefert werden. Somit ist das Testen von Touch und Tags etc. nicht möglich.	100	0	0	Keine, nicht möglich.	Alternative Testmethoden nutzen, um Testumgebung zu simulieren.	1	2
2	Fehlein-schätzung des Zeitaufwands	Der Aufwand wurde falsch eingeschätzt. Somit stimmt der Projektplan nicht.	0	20	0	Kontinuierliche Überprüfung der Projektplanung und eventuelle Anpassung. Verantwortung für aktuellen Projektplan an einem Teammitglied zuweisen.	Konzentration auf die wichtigsten Kernfunktionalitäten, ev. werden gewisse Funktionen weggelassen.	1	6
3	Einarbeitung Surface	Die Einarbeitung in Surface benötigt mehr Zeit als ursprünglich geplant.	0	50	0	Michael Gfeller hat bereits Erfahrung mit Surface. Bei allfälligen Problemen kann und soll er um Rat angefragt werden.	Der Funktionsumfang des Projektes wird wenn nötig reduziert.	1	4

PDF auf Surface darstellen 4 kompliziert	Es ist kompliziert, ein PDF auf dem Surface darzustellen. Möglicherweise muss das PDF Dokument in ein anderes Format umgewandelt werden.	0	50	Das Problem wird früh angegangen und bereits im Architekturprototypen umgesetzt.	Umwandlung in anderen Dokumenttypen wird programmiert oder externe Hilfe wird angefordert.	1	1
Sharepoint 5 Anbindung	Der Cronjob kann nicht am Sharepoint angebunden werden. Deshalb müssten die PNs manuell eingetragen werden, was zusätzliche variable Kosten verursacht.	5	0	Es wird ein klares, einfaches API Interface vorgegeben (falls möglich sogar inkl. Unit Tests), damit es möglichst einfach ist, die Anbindung zu programmieren.	Falls die Anbindung nicht möglich sein sollte, muss für den Betrieb ein GUI Interface programmiert werden, das einfach bedient werden kann. Somit bleiben die variablen Kosten tief.	2	
Papierprototyp benötigt mehrere 6 Anpassungen	Der Papierprototyp der zu erstellenden Anwendung ist für Benutzer nicht gut verständlich. Daher müssen Anpassungen vorgenommen werden.	0	12	Vor der Erstellung des Papierprototyps werden in einem Creative Workshop verschiedenste Varianten für die Anwendung ausgearbeitet und 0 miteinander verglichen.	Die nötigen Anpassungen werden vorgenommen und der Papierprototyp wird erneut getestet.	2	2

<p>Aus jedem XPS soll ein Bild für jede PN extrahiert werden. Dies muss automatisch ablaufen.</p> <p>Schadenspotenzial: Pro PN ca 10 Minuten, um Bild manuell zu extrahieren + höhere Betriebskosten.</p>								
Bild aus XPS 7 extrahieren		0	100	Möglichst früh umsetzen (noch in SP3), um früh reagieren zu können.	Bilder manuell extrahieren. Kostet ca. 50 h + höhere Betriebskosten.	3	3	
Total			232	0				

User Stories Project Flip 2.0, Zühlke, 20.12.2011															Scrum Poker		
Epic	Titel	User Story Als Surface Benutzer möchte ich...	Definition of done	Abgenommen durch ... am ...	Priorit	Schätzung in Tagen	Sprint	Identified	Comment	Kategorie	Redmine Ticket #	Schätzung in Stunden	DT	CH	LE		
Übersicht	Übersicht für PN	die PN in einer Übersicht anzeigen lassen, damit ich mir einen Überblick über die verschiedenen PN verschaffen kann.	Alle PNs werden in einem Gitter dargestellt.	Christian, Markus, 31.10.2011	1,01	Must	3	3	14.10.2011 Dazu gehört auch die Hintergrundfunktional also z.B. PN laden, ...	Basic	574	24	1	2	3		
Detailansicht	Detailansicht PN	eine Detailansicht der PN sehen, damit ich die einzelnen PN lesen kann und genauere Infos erhalte.	Eine bestimmte PN wird in einem separaten Container dargestellt, wobei die XPS Datei geöffnet sein soll.	Christian, Markus, 31.10.2011	1,02	Must	2	3	14.10.2011	Basic	575	16	2	2	2		
Übersicht	Navigation "Übersicht -> Detail"	von der Übersicht über alle PN in eine Einzelansicht wechseln, damit ich die einzelnen PN lesen kann und genauere Infos erhalte.	Durch das Anklicken einer PN in der Übersicht wird die Detailansicht mit genau dieser PN geöffnet.	Christian, Markus, 31.10.2011	1,03	Must	1	3	14.10.2011 Achtung: MVVM ;)	Navigation	576	8	0,5	0,5	1		
Detailansicht	Navigation "Detail -> Übersicht"	von der Detailansicht zur Übersicht navigieren, damit ich mir einen Überblick über die aktuelle Detailansicht verschaffen kann.	Durch das Anklicken eines zurück-Buttons wird die Detailansicht geschlossen und die Übersicht wird wieder angezeigt.	Christian, Markus, 31.10.2011	1,04	Must	1	3	14.10.2011 Achtung: MVVM ;) Bilder automatisch extrahieren: stellt Risiko dar, deshalb SP3	Navigation	577	8	0,5	0,5	1		
Übersicht	Darstellung der verkleinerten PN mit Bild	die PN mithilfe eines Bildes unterscheiden, wenn sie verkleinert sind, um einfacher eine Auswahl zu treffen.	Die PNs werden in einem Gitter mit je einem Bild dargestellt.	Christian, Markus, 31.10.2011	3	Must	2	3	14.10.2011	Extra	578	16	1	3	2		
Detailansicht	Navigation "Detailansicht -> Detailansicht"	bei der Detailansicht einer PN zur nächsten oder vorherigen PN wechseln, damit ich alle PN nacheinander anschauen kann.	In der Detailansicht kann auf einen Button "nach rechts" oder "nach links" geklickt werden, wobei die aktuelle PN mit der rechten bzw. linken PN ausgetauscht wird.	Christian, Markus, 31.10.2011	1,05	Must	1	3	14.10.2011	Navigation	582	8	1	0,5	1		
Detailansicht	Tags zu PN angezeigt	zu einer PN zusätzliche Begriffe anzeigen, damit ich die PN einem bestimmten Gebiet zuordnen kann.	Tags, die zu einer PN gehören, werden in der Detailansicht zur PN angezeigt.	Christian, Markus, 14.11.2011, Christian, 25.11.2011	3,001	Must	2	4	14.10.2011	Plus	586	16	1	2	2		
Filtern	Tags aggregiert	bei den Tags statt vielen spezifischen Tags eine kleinere Auswahl von aggregierten Tags sehen, um die Übersicht zu behalten und nicht zu viele Elemente auf dem Bildschirm zu sehen. (Bsp: Java ME, Java EE, Java Beans -> wird aggregiert zu Java)	Die Tags werden aggregiert angezeigt.	Markus, 14.11.2011, Christian, 25.11.2011	5	Shoul	1	4	23.10.2011 Gemäss Sitzungsprotokoll 17.10.11:	Plus	590	8	1	1	1		
Filtern	Aggregierte Tags anpassbar	Als Surface Admin möchte ich die Zuordnungen der Tags bearbeiten können, damit ich neue Tags in eine Oberkategorie aggregieren kann und damit ich die aggregierten Elemente ändern kann.	Config File kann von einem Admin bearbeitet werden.	Markus, 14.11.2011, Christian, 25.11.2011	5	Shoul	1	4	31.10.2011 Sitzungsprotokoll 31.10.11: So sinnvoll konfigurierbar und anpassbar für die spätere Verwendung sein. Funktionalität steht im Vordergrund. Speicherung als csv, Textfile oder xml. Eher einfach maschinenlesbares Format, xml bevorzugt.	Plus	591	8	1	1	0,5		
Filtern	Filter in Übersicht setzen	einen Filter in der Übersicht setzen, damit die angezeigten PN eingeschränkt werden.	In der Übersicht kann ein Filter gesetzt werden, sodass alle PN nach dem gesetzten Filter gefiltert werden. Die tatsächliche Auswahl des Filters ist hier nicht inbegriffen.	Markus, 14.11.2011, Christian, 25.11.2011	3,01	Must	1	4	14.10.2011	Plus	587	8	1	1	0,5		
Filtern	Filterkriterium auswählen	das Filterkriterium aus einer nach Kategorien gruppierten Liste auswählen können, damit ich den gesuchten Begriff einfach finden kann.	Ein Filterkriterium kann aus einer Menge von Filterkriterien ausgewählt werden.	Markus, 14.11.2011, Christian, 25.11.2011	3,015	Must	2	4	14.10.2011	Plus	588	16	1	2	2		
Filtern	Filter in Übersicht entfernen	einen Filter in der Übersicht entfernen, damit mir mehr PN angezeigt werden.	Ein zuvor gewähltes Filterkriterium kann wieder entfernt werden, wobei die PN nicht mehr nach diesem Filterkriterium gefiltert werden.	Markus, 14.11.2011, Christian, 25.11.2011	3,02	Must	1	4	14.10.2011	Plus	589	8	1	1	0,5		

User Stories Project Flip 2.0, Zühlke, 20.12.2011														Scrum Poker		
Epic	Titel	User Story Als Surface Benutzer möchte ich...	Definition of done	Abgenommen durch ... am ...	Priorit		Schätzung in Tagen	Sprint	Identified	Comment	Kategorie	Redmine Ticket #	Schätzung in Stunden	DT	CH	LE
Filtern	Filter in Detailansicht entfernen	einen Filter in der Detailansicht entfernen, damit ich dazu nicht zuerst zur Übersicht wechseln muss.	In der Detailansicht kann der Filter entfernt werden.	Markus, 14.11.2011, Christian, 25.11.2011	3,04	Must	0	4	14.10.2011		Plus	589	0	0,5	0,5	0,5
Design	Externes Design festgelegt und validiert	Als Entwickler möchte ich für die Design User Stories eine "Definition of Done" festlegen können, damit der Abschluss der User Stories validiert werden kann.	Das externe Design wurde im Photoshop erstellt und dem Kunden gezeigt, der damit zufrieden war.	Christian, Markus, 11.11.2011	5	Could	2,5	4	31.10.2011		Extra	596	20	-	2,5	-
Filtern	Filter in Detailansicht setzen	einen Filter in der Detailansicht setzen, damit ich dazu nicht zuerst zur Übersicht wechseln muss.	In der Detailansicht kann ein Filter gesetzt werden, wobei nach dem Setzen des Filters zur Übersicht gewechselt wird.	Markus, 28.11.2011, Christian, 09.12.2011	3,03	Must	0,5	5	14.10.2011		Plus	601	4	0,5	0,5	0,5
Detailansicht	Filtern nach mehreren Kriterien	nach mehreren bestimmten Kriterien filtern, um eine begrenzte Auswahl zu erhalten.	Es können mehrere Filter gleichzeitig eingegeben werden, wobei nach allen gefiltert wird (AND).	Markus, 28.11.2011, Christian, 09.12.2011	5,001	Could	1	5	14.10.2011		Extra	602	8	0,5	2	1
Detailansicht	Animierte Navigation "Detail -> Übersicht"	in der Detailansicht erkennen können (z. B. durch Animation), damit ich den Weg von der Detailansicht zur Übersicht finde.	Der abgedunkelte Hintergrund kann angeklickt werden, wodurch zur Übersicht gewechselt wird.	Markus, 28.11.2011, Christian, 09.12.2011	5,01	Could	2	5	14.10.2011		Animation	603	16	2	1	2
Übersicht	Scrollerkennung in Übersicht dargestellt	an der Darstellung der Liste erkennen können, dass durch die PN gescrollt werden kann.	In der Übersicht existiert ein grafisches Element, welches auf mehr Project Notes hindeutet.	Markus, 28.11.2011, Christian, 09.12.2011	5,03	Could	0,5	5	14.10.2011		Animation	604	4	0,5	0,5	0,5
Detailansicht	Schöne Darstellung der Tags	die zu einer PN zugehörigen Begriffe schön darstellen, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht.	Die Tags werden gemäss externem Design dargestellt.	Markus, 28.11.2011, Christian, 09.12.2011	5,05	Could	1	5	14.10.2011	Ohne optimale Platzausnutzung	Extra	606	8	1	1	1
Filtern	Schöne Darstellung des Filters	den Filter schön darstellen, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht.	Der Filter wird gemäss dem externen Design dargestellt	Markus, 28.11.2011, Christian, 09.12.2011	7	Could	1	5	14.10.2011		Extra	608	8	1	0,5	1
Detailansicht	Lesemodus PN anzeigen	eine PN vergrössern können und einen abgedunkelten Hintergrund sehen, damit ich sie besser lesen kann.	Die PN kann so weit verbreitert werden, wie neben dem Filter und den Tags noch Platz ist.	Christian, Markus, 09.12.2011	5,05	Could	1	6	14.10.2011		Plus	605	8	1	1	1
Detailansicht	Animation für Navigation "Detailansicht -> Detailansicht"	eine Animation sehen, wenn ich durch die PNs navigiere, damit es für mich besser ersichtlich ist, dass die Project Note gewechselt hat.	Es existiert eine Animation, wenn von der einen Detailansicht in die andere Detailansicht gewechselt wird.	Christian, Markus, 09.12.2011	7	Could	1	6	14.10.2011	Kommt darauf an wie komplex die Animation ist.	Animation	607	8	0,5	1	2
Übersicht	Anzahl PN bei Übersicht anzeigen	die Anzahl PN in der Übersicht sehen, damit ich weiss, wie gross die Liste ist.	Oberhalb des Filters wird dargestellt, wie viele von der maximalen Anzahl an Project Notes aktuell in der Übersicht angezeigt werden.	Christian, Markus, 09.12.2011	7	Could	0,5	6	14.10.2011	Bsp.: 39 von 340	Animation	615	4	0	0,5	0,5
Sonstiges	Easteregg	Als Entwickler oder Zühlke Mitarbeiter möchte ich ein Easteregg ausführen können, damit ich Spass haben und Kunden beeindrucken kann.	Info View für Beteiligte Parteien & Personen erscheint beim Klicken des Info Buttons.	Christian, Markus, 09.12.2011	11	Future	0,5	6	14.10.2011	Info View / About	Extra	629	4	?	?	2
Filtern	Animation Filter -> Übersicht	eine Animation sehen, wenn die Ansicht geschlossen und der Filter gesetzt wird, damit ich sehe, dass und wo der Filter nun gesetzt ist. (Filter highlighten, nach dem setzen des Kriteriums, gemäss Protokoll)	Wenn ein Filterkriterium gesetzt wird, so wird es eine Animation ausgelöst.	Christian, Markus, 09.12.2011	9	Could	1	6	14.10.2011		Animation	616	8	0,5	1	1
Filtern	Animation Filterkriterium	bei der Auswahl einer Kategorie die Unterkategorie animiert eingeblendet sehen, damit mir die Applikation Spass macht.			5	Shoul	1	Future	14.10.2011		Animation		8	1	1	1
Sonstiges	PN ausgedruckt	eine PN ausdrucken können, um sie mitzunehmen und später genau zu lesen.			10	Future	2	Future	14.10.2011	Ohne Surface schlecht testbar Ohne Surface nicht testbar. Kommt darauf an, ob mit	Extra		16	1	2	2
Sonstiges	Badge erkennen	einen Besucher Badge auf den Surface legen können, der dann automatisch erkannt wird, um die zugehörigen Tags zu sehen.			10	Future	3	Future	14.10.2011	OCR oder Scan-Tag	Extra		24	3	3	5

User Stories Project Flip 2.0, Zühlke, 20.12.2011																
		User Story											Scrum Poker			
Epic	Titel	Als Surface Benutzer möchte ich...	Definition of done	Abgenommen durch ... am ...	Priorit	Schätzung in Tagen	Sprint	Identified	Comment	Kategorie	Redmine Ticket #	Schätzung in Stunden	DT	CH	LE	
Sonstiges	PN verschickt	eine PN an eine Emailadresse schicken können, um sie später genau lesen zu können.			10	Future	3	Future	14.10.2011	Ohne Surface schlecht testbar	Extra	24	2	3	3	
Sonstiges	Animation für aktionslose Beschriftung	beim Anklicken der aktionslosen Beschriftung bei den Tags ein Feedback erhalten, damit ich merke, dass keine Aktion hinter der Beschriftung steht.			9	Could	0,5	Unplanne	14.10.2011	Bringt Reaktion auf nicht klickbare Elemente etwas? Ausprobieren und daher umsetzen.	Animation	4	1	0,5	0,5	
Filtern	Volltextsuche	mithilfe einer Volltextsuche PN filtern können, damit ich nur diese anzeigen lassen kann, die mich wirklich interessieren.			7	Could	1	Unplanne	23.10.2011		Plus	8	1	1	1	
Demomodus	Demomodus erstellt	durch einen Demomodus auf die Applikation aufmerksam gemacht werden, damit ich animiert werde, den Surface zu bedienen.			9	Future	2	Unplanne	14.10.2011		Plus	16	?	2	2	
Demomodus	Demomodus erkennen und verlassen	den Demo Modus erkennen, damit ich diesen verlassen kann, um selbst zu navigieren.			9	Future	1	Unplanne	14.10.2011		Plus	8	0,5	1	1	
Detailansicht	Animation für Lesemodus	eine Animation sehen, wenn ich in den Lesemodus wechsele, damit mir die Applikation mehr Spass macht.			11	Future	1	Unplanne	14.10.2011		Animation	8	1	0,5	1	
Detailansicht	Platzsparende Darstellung der Tags	die zu einer PN zugehörigen Begriffe platzsparend darstellen, damit mir mehr Informationen gleichzeitig angezeigt werden.			11	Future	2	Unplanne	14.10.2011	Platzausnutzung optimal gestalten	Extra	16	-	-	2	
Übersicht	Animation Übersicht -> Filter	eine Animation sehen, wenn die Kriterienliste für das Filtern geöffnet wird, damit ich den Filtermodus erkenne.			11	Future	1	Unplanne	14.10.2011		Animation	8	0,5	1	1	
Filtern	Kontextmenu bei Tag antippen	beim Antippen eines Tags ein Kontextmenu für weiteres Filtern anzeigen, damit ich entweder alle PN zu einem Thema anzeigen kann oder die aktuelle Auswahl weiter einschränken kann.			11	Future	0,5	Unplanne	14.10.2011	Gemäss Sitzungsprotokoll: Kontextmenu bei Tag nötig? Oder auf andere Weise umsetzen? Das welches am ehesten erwartet wird?	Plus	4	2	0,5	0,5	
Übersicht	Weisser Rand bei Bild erkannt und so ausgeschnitten	bei der Miniaturansicht einer PN mit zugehörigem Bild das Bild in möglichst grosser Form darstellen (grosse Bilder der PN auch im Querformat), damit ich die PN besser erkennen kann.			11	Future	1	Unplanne	23.10.2011		Extra	8	0,5	1	1	
Total							49,5					396	33,5	45	50,5	

Project Flip 2.0 Zühlke - Report 1

#	Zielversion	Thema	Kategorie	Geschätzter Aufwand	Aufgewendete Zeit	Beginn	Abgabedatum
SP7 (21)							
636	SP7	Dokument "Realisierung & Test" geschrieben		3.0	5.0	13.12.2011	13.12.2011
635	SP7	Usability Tests durchgeführt und protokolliert		4.0	3.5	13.12.2011	13.12.2011
640	SP7	Codereview 2 durchgeführt, dokumentiert, implementiert		4.0	4.0	16.12.2011	16.12.2011
634	SP7	SE Dok: "Entwurf" geschrieben		16.0	10.0	13.12.2011	19.12.2011
633	SP7	Code dokumentiert		5.0	2.0	12.12.2011	19.12.2011
632	SP7	Weiterentwicklung dokumentiert		8.0	5.25	12.12.2011	19.12.2011
475	SP7	Meeting, 19.12.11		3.0	4.75	19.12.2011	19.12.2011
630	SP7	Abstract geschrieben		3.0	1.5	20.12.2011	20.12.2011
641	SP7	Allgemeine Korrekturen, kleine Anpassungen		19.0	16.5	18.12.2011	23.12.2011
639	SP7	Einleitung Technischer Bericht geschrieben		3.0	2.0	15.12.2011	23.12.2011
631	SP7	Dokumente zusammenfügen, PDF generieren		10.0	14.5	12.12.2011	23.12.2011
562	SP7	Video und Wiki Seite erstellt		8.0	14.0	12.12.2011	23.12.2011
561	SP7	Tools sind beschrieben		1.0	1.0	12.12.2011	23.12.2011
554	SP7	Codereview dokumentiert		4.0	1.0	12.12.2011	23.12.2011
552	SP7	Installationsanleitung geschrieben		2.0	1.0	12.12.2011	23.12.2011
548	SP7	Extended Management Summary geschrieben		14.0	10.5	12.12.2011	23.12.2011
547	SP7	Management Summary geschrieben		0.0	0	12.12.2011	23.12.2011
545	SP7	Persönlicher Bericht geschrieben		12.0	11.5	12.12.2011	23.12.2011
544	SP7	CD gebrannt und abgegeben		3.0	3.0	12.12.2011	23.12.2011
520	SP7	Risikomanagement nachgeführt		1.0	0.5	12.12.2011	23.12.2011
638	SP7	Codereview Verbesserungen implementiert		4.0	15.0	24.12.2011	26.12.2011
SP6 (25)							
471	SP6	Meeting, 05.12.11		6.0	6.25	05.12.2011	05.12.2011
473	SP6	Meeting, 09.12.11		6.0	5.5	09.12.2011	09.12.2011
637	SP6	SVN Tag SP6 erstellt		0.5	0.5	12.12.2011	12.12.2011
629	SP6	Info View / Easteregg	User Story	4.0	8.0	08.12.2011	12.12.2011
628	SP6	Codereview durchgeführt		3.0	3.0	09.12.2011	12.12.2011
623	SP6	Verbesserungen aus Usability Test gemacht		8.0	19.0	05.12.2011	12.12.2011
622	SP6	Dokumente vervollständigt / diverse kleine Anpassungen		4.0	4.0	05.12.2011	12.12.2011
621	SP6	Refactoring		5.0	9.0	29.11.2011	12.12.2011

620	SP6	Betriebskonzept / Installationsanleitung erarbeitet		4.0	1.5	28.11.2011	12.12.2011
619	SP6	Funktionale Anforderungen dokumentiert		2.0	1.25	28.11.2011	12.12.2011
617	SP6	3-D Darstellung der PNs dokumentiert		4.0	5.5	28.11.2011	12.12.2011
616	SP6	Animation Filter -> Übersicht	User Story	8.0	10.0	28.11.2011	12.12.2011
615	SP6	Anzahl PN bei Übersicht angezeigt	User Story	4.0	5.5	28.11.2011	12.12.2011
613	SP6	Dokument "Realisierung & Test" geschrieben		3.0	3.0	28.11.2011	12.12.2011
612	SP6	SE Dok: "Entwurf" geschrieben		6.0	7.5	28.11.2011	12.12.2011
564	SP6	Usability Tests durchgeführt und protokolliert		12.0	8.75	02.12.2011	12.12.2011
559	SP6	Tests dokumentiert		3.0	0.5	28.11.2011	12.12.2011
551	SP6	Aufgabenstellung unterschrieben		1.0	0.75	28.11.2011	12.12.2011
550	SP6	Lizenzvereinbarung unterschrieben		4.0	2.0	28.11.2011	12.12.2011
549	SP6	Erklärung eigenständige Arbeit unterschrieben		1.0	1.0	28.11.2011	12.12.2011
546	SP6	Abstract geschrieben		9.5	9.75	12.12.2011	12.12.2011
518	SP6	Risikomanagement nachgeführt		1.0	0.5	28.11.2011	12.12.2011
517	SP6	Sprint 7 geplant		4.0	0.5	09.12.2011	12.12.2011
474	SP6	Meeting, 12.12.11		3.0	4.75	12.12.2011	12.12.2011
614	SP6	Lesemodus PN anzeigen	User Story	4.0	9.25	29.11.2011	13.12.2011
SP5 (14)							
604	SP5	Scrollerkennung in Übersicht dargestellt	User Story	4.0	29.0	14.11.2011	21.11.2011
600	SP5	Meeting, 25.11.11		4.5	9.0	25.11.2011	25.11.2011
462	SP5	Sprint 6 geplant		6.0	1.0	25.11.2011	25.11.2011
618	SP5	Anforderungen und Domain Modell Dokumente korrigiert		3.0	3.5	28.11.2011	28.11.2011
608	SP5	Schöne Darstellung des Filters	User Story	8.0	10.5	14.11.2011	28.11.2011
607	SP5	Animation für Navigation "Detailansicht -> Detailansicht"	User Story	8.0	19.75	14.11.2011	28.11.2011
606	SP5	Schöne Darstellung der Tags	User Story	8.0	7.5	14.11.2011	28.11.2011
605	SP5	Lesemodus PN anzeigen	User Story	4.0	12.0	14.11.2011	28.11.2011
603	SP5	Animierte Navigation "Detail -> Übersicht"	User Story	16.0	6.25	14.11.2011	28.11.2011
602	SP5	Filtern nach mehreren Kriterien	User Story	8.0	9.0	14.11.2011	28.11.2011
601	SP5	Filter in Detailansicht setzen	User Story	4.0	3.5	14.11.2011	28.11.2011
558	SP5	Tests dokumentiert		3.0	3.0	14.11.2011	28.11.2011
488	SP5	Risikomanagement nachgeführt		1.0	1.0	14.11.2011	28.11.2011
472	SP5	Meeting, 28.11.11		3.0	5.25	28.11.2011	28.11.2011
SP4 (19)							
594	SP4	Padding / Margin in Übersicht-Buttons entfernen		3.0	3.0	04.11.2011	04.11.2011
584	SP4	Meeting, 04.11.11		6.0	9.0	04.11.2011	04.11.2011

567	SP4	NF Anforderungen dokumentiert		1.5	2.0	01.11.2011	04.11.2011
469	SP4	Meeting, 07.11.11		3.0	3.25	07.11.2011	07.11.2011
597	SP4	Navigation "Detailansicht -> Detailansicht"		2.0	3.5	07.11.2011	08.11.2011
470	SP4	Meeting, 11.11.11		6.0	6.5	11.11.2011	11.11.2011
459	SP4	Sprint 5 geplant		6.0	0.5	11.11.2011	11.11.2011
596	SP4	Layout definiert		16.0	22.25	06.11.2011	14.11.2011
593	SP4	Refactoring		6.0	8.5	31.10.2011	14.11.2011
591	SP4	Aggregierte Tags anpassbar	User Story	8.0	7.5	31.10.2011	14.11.2011
590	SP4	Tags aggregieren	User Story	8.0	10.0	31.10.2011	14.11.2011
589	SP4	Filter in Übersicht entfernen	User Story	8.0	4.25	31.10.2011	14.11.2011
588	SP4	Filterkriterium auswählen	User Story	16.0	11.5	31.10.2011	14.11.2011
587	SP4	Filter in Übersicht setzen	User Story	8.0	10.5	31.10.2011	14.11.2011
586	SP4	Tags zu PN angezeigt	User Story	16.0	8.0	31.10.2011	14.11.2011
585	SP4	PivotViewer untersucht und dokumentiert		2.0	3.5	31.10.2011	14.11.2011
557	SP4	Tests dokumentiert		3.0	2.5	31.10.2011	14.11.2011
487	SP4	Risikomanagement nachgeführt		1.0	0.5	31.10.2011	14.11.2011
448	SP4	Meeting, 14.11.11		6.0	3.0	14.11.2011	14.11.2011
SP3 (22)							
571	SP3	Projektplan und User Stories angepasst		4.0	6.0	17.10.2011	23.10.2011
566	SP3	Guidelines gelesen / überflogen / dokumentiert		4.0	8.0	18.10.2011	24.10.2011
468	SP3	Meeting, 24.10.11		3.0	3.25	24.10.2011	24.10.2011
592	SP3	NF Anforderungen dokumentiert		4.0	2.5	17.10.2011	31.10.2011
583	SP3	SVN Tag SP3 erstellen		0.25	0.25	31.10.2011	31.10.2011
582	SP3	Navigation "Detailansicht -> Detailansicht"	User Story	8.0	5.75	17.10.2011	31.10.2011
580	SP3	PN hineinladen stabiler / fehlertoleranter gestaltet		8.0	7.25	24.10.2011	31.10.2011
579	SP3	UnitTests für Prototypen geschrieben		8.0	9.5	17.10.2011	31.10.2011
578	SP3	Darstellung der verkleinerten PN mit Bild	User Story	16.0	0	17.10.2011	31.10.2011
577	SP3	Navigation "Detail -> Übersicht"	User Story	8.0	6.25	17.10.2011	31.10.2011
576	SP3	Navigation "Übersicht -> Detail"	User Story	8.0	6.5	17.10.2011	31.10.2011
575	SP3	Detailansicht PN	User Story	16.0	10.5	17.10.2011	31.10.2011
574	SP3	Übersicht für PN	User Story	24.0	0	17.10.2011	31.10.2011
573	SP3	Anzeige der PNs in Übersicht		8.0	7.75	17.10.2011	31.10.2011
572	SP3	Bilder aus XPS extrahiert		16.0	8.5	17.10.2011	31.10.2011
569	SP3	Diskussion Accessibility dokumentiert		2.0	1.25	17.10.2011	31.10.2011
568	SP3	Design Constraints dokumentiert		3.0	0.75	17.10.2011	31.10.2011

556	SP3	Tests dokumentiert		6.0	3.0	17.10.2011	31.10.2011
515	SP3	Domain Model erstellt		4.0	4.75	17.10.2011	31.10.2011
486	SP3	Risikomanagement nachgeführt		1.0	0.75	17.10.2011	31.10.2011
458	SP3	Sprint 4 geplant		6.0	3.5	31.10.2011	31.10.2011
447	SP3	Meeting, 31.10.11		6.0	7.5	31.10.2011	31.10.2011
SP2 (16)							
509	SP2	Creative Workshop durchgeführt		12.0	12.0	03.10.2011	07.10.2011
446	SP2	Meeting, 10.10.11		6.0	6.0	10.10.2011	10.10.2011
570	SP2	Sprint 7 vorgeplant		2.0	2.0	03.10.2011	17.10.2011
563	SP2	Guidelines gelesen / überflogen / dokumentiert		2.0	2.0	03.10.2011	17.10.2011
543	SP2	Creative Workshop dokumentiert		6.0	15.0	03.10.2011	17.10.2011
514	SP2	User Stories priorisiert		9.0	4.5	03.10.2011	17.10.2011
513	SP2	Scrum Poker durchgeführt		3.0	3.0	03.10.2011	17.10.2011
512	SP2	User Stories definiert		6.0	6.0	03.10.2011	17.10.2011
511	SP2	Papierprototyp getestet und ausgewertet		8.0	20.0	03.10.2011	17.10.2011
510	SP2	Papierprototyp erstellt		8.0	15.5	03.10.2011	17.10.2011
508	SP2	Szenarien erstellt		12.0	6.5	03.10.2011	17.10.2011
507	SP2	Personas erstellt		3.0	2.0	03.10.2011	17.10.2011
506	SP2	Vision erstellt		2.0	2.0	03.10.2011	17.10.2011
485	SP2	Risikomanagement nachgeführt		1.0	2.5	03.10.2011	17.10.2011
467	SP2	Meeting, 17.10.11		6.0	7.5	17.10.2011	17.10.2011
457	SP2	Sprint 3 geplant		6.0	4.5	14.10.2011	17.10.2011
SP1 (23)							
456	SP1	Planung Sprint 1		6.5	6.5	19.09.2011	19.09.2011
445	SP1	Meeting, 19.09.11		6.0	6.0	19.09.2011	19.09.2011
465	SP1	Entwicklungsumgebung installieren/einrichten		6.0	6.0	19.09.2011	20.09.2011
464	SP1	Dokumenterstellung		4.0	6.0	19.09.2011	23.09.2011
449	SP1	PDF zu XPS konvertieren		8.0	7.0	23.09.2011	23.09.2011
494	SP1	Brainstorming, Interview analysieren		7.5	7.5	26.09.2011	26.09.2011
480	SP1	Interviews durchführen		9.0	3.0	26.09.2011	26.09.2011
463	SP1	Meeting, 26.09.11		3.0	6.0	26.09.2011	26.09.2011
519	SP1	Planung Sprint		10.0	10.0	30.09.2011	30.09.2011
496	SP1	Einarbeitung Scrum		8.0	6.0	26.09.2011	30.09.2011
491	SP1	Sitzungsprotokolle		1.0	1.0	26.09.2011	30.09.2011
482	SP1	XPS auf Surface darstellen		3.0	7.0	21.09.2011	30.09.2011

481	SP1	Interviews erstellen & organisieren		2.0	1.0	21.09.2011	30.09.2011
453	SP1	Szenarien		1.0	1.0	23.09.2011	30.09.2011
452	SP1	Personas		2.0	2.5	23.09.2011	30.09.2011
492	SP1	Studium Technologie		4.0	4.5	19.09.2011	03.10.2011
490	SP1	Flip 1.0 analysieren und dokumentieren		4.0	3.25	26.09.2011	03.10.2011
489	SP1	Projektmanagement		4.0	6.5	19.09.2011	03.10.2011
484	SP1	Risiko Management		5.0	1.0	19.09.2011	03.10.2011
479	SP1	Administrative Arbeiten		2.0	2.0	19.09.2011	03.10.2011
466	SP1	Meeting, 03.10.11		12.0	13.5	03.10.2011	03.10.2011
454	SP1	Architekturprototyp		20.0	15.25	19.09.2011	03.10.2011
450	SP1	Vision		3.0	4.0	19.09.2011	03.10.2011
SP0 (3)							
460	SP0	Kickoff Meeting		7.5	7.5	31.08.2011	31.08.2011
455	SP0	Planung Sprint		2.5	2.5	31.08.2011	31.08.2011
444	SP0	WPF Einführung		9.0	9.0	07.09.2011	07.09.2011

Pivot View of Netflix Instant Watch Movies





Microsoft Surface 2.0

Design and Interaction Guide

Principles & Guidelines for Designing and Developing
Surface Applications

July 2011

COPYRIGHT

This document is provided for informational purposes only, and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results from the use of this document remains with the user. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, financial and other data, and events depicted herein, are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, places, financial or other data, or events is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, Microsoft Surface, the Microsoft Surface logo, and Windows are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

INTRODUCTION

Microsoft Surface enables developers and designers to deliver amazing, social, and highly interactive experiences to their customers. People can use the 360° degree interface from all sides for face-to-face collaboration, cooperation, and building trust. It's intimate enough to facilitate the close back-and-forth of a consultation and makes intuitive sense to people, young and old. Touch. Push. Pull. Turn. Place an object on the screen. Surface creates new opportunities to socially interact with people and content.

Simply put, there is no easier way to bring people together to connect, learn, have fun, and decide. Surface is what computing always promises to be – intuitive, informative, engaging, and helpful – and with it designers and developers create experiences customers will seek out and remember.

Developing compelling Surface experiences requires a different approach to interface design. This document presents design principles and guidelines to address key aspects of application interface design including: interaction, visual, sound, text, and more. Use these principles and practices as a starting point to get the most out of the Surface software and hardware platform's unique capabilities.

Section 2.0:

GETTING STARTED

The following section highlights important things to keep in mind while planning your Surface applications.

Section 2.1:

SURFACE BRAND TENETS

Delightful

Enchanting, surprising, rewarding

Surface rewards people's curiosity in a responsive and captivating way that connects with their senses of touch, sight, and hearing.

Smart

Intuitive, savvy, perceptive

Surface anticipates people's next move. It entices them and reveals key information right when it's needed.

Understated

Elegant, subtle, intriguing

Surface helps people perceive the potential contained within the experience through a thoughtful and tactful approach.

Section 2.2:

KEY CONSIDERATIONS

1. Pixels versus inches

Microsoft Surface hardware exposes the

`Microsoft.Surface.Input.InteractiveSurfaceDevice.PhysicalScaleTransform` property.

This property represents the ratio between 96 dot per inch (DPI) and the logical DPI used by Microsoft Surface units. Using this transform, 96 pixels will result in one real inch on Microsoft Surface displays.

The `LogicalDpiX` and `LogicalDpiY` properties are also exposed. These return the number of pixels that show as one inch on a Surface device.

These properties allow people to write applications where a `UIElement` will have the same physical display size, no matter what type of Surface device is used. These properties help design applications that have content that is properly sized, no matter the screen dimension or resolution. For instance, a keyboard should come up the same size no matter what hardware is used.

Disclaimer: Not all displays expose the real size of a pixel, in that case the default 96 DPI is assumed for that type of hardware.

2. Globalization and localization

Microsoft Surface applications are available in a number of languages and regions around the world. Developers selling their applications to global markets should ensure their applications are world ready. Follow best practices designing the application interface to support varying text string lengths, date formats; and be aware of cultural sensitivities around use of colors, images, and geopolitical issues. MSDN online has a variety of topics dealing with these best practices.

Generally, you should provide at least 40% buffer space for localized strings.

3. Understanding your audience

Prior to Surface application development, it is critical that application developers and designers understand who they are developing for, what setting the application is likely to be used, and the business goals for the application.

Section 3.0:

DESIGN PRINCIPLES

Design is a critical part of the Microsoft Surface experience. Surface applications are based on a special set of principles that helps ensure content is always at the heart of the experience.

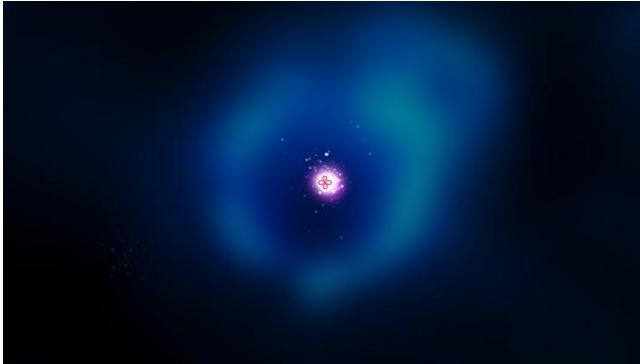
Customer requirements and implementations will vary across applications, so it's strongly recommended that developers use these principles as the foundation for designing and stylizing their applications. Surface design principles should be used as the inspirational

starting point to create consistent and enjoyable experiences that draw people in. Surface application design follows five principles:

01 Simple

Light and open

The Surface experience is never overly decorated or embellished.

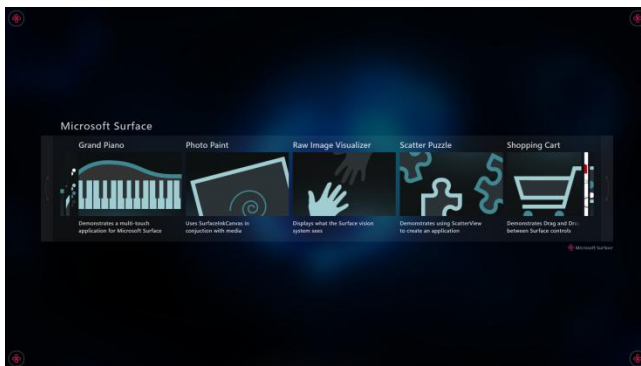


The Begin Experience icon is a simple and straightforward navigation point to begin the Surface experience. In contrast with the playful, organic Attract mode, the Begin icon is graphic and minimal to indicate its function as a navigation point.

02 Organized

Hierarchical and structured

The most important elements have the most visual weight.

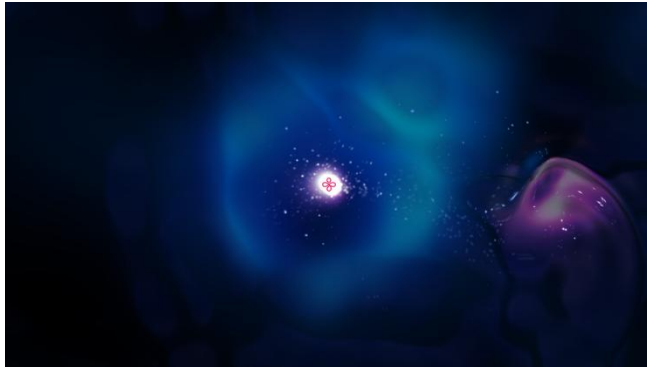


The Surface Launcher is visually organized and structured in a way that places focus on the application icons, titles, and descriptive text. The Launcher container is subtle, and helps maintain a sense of organization when placed over any number of backgrounds.

03 Authentically Digital

True-to-form, honest

Surface uses an honest and direct design style that reflects the digital medium and emulates the physical world only when appropriate.



The Attract state is based on real world physics and water interaction. It adds stylized digital particles and colorful reflections to move beyond what is normally possible in the real world, making the experience true to the digital medium.

04 Content Oriented

Focused and discoverable

Content is always at the core of the experience – controls and chrome are secondary.



The photo content shown here is the most important part of the experience. The Element Menu indicator appears only if a photo is touched. People can then touch the Element Menu icon to reveal a number of interaction options.

05 Lively

Soulful and responsive

Surface is always responsive – animation and motion enhance content and bring it to life onscreen.



Surface transitions bring the experience to life. The application loading transition shown here illustrates the way Surface responds to launching an application; the application icon scales slightly up under the user's finger, the Launcher scales back and dims, and the loading bar appears under the icon to indicate loading progress.

Section 3.1:

INTERACTION DESIGN GUIDELINES

Interaction design defines the interplay of the software experience with people's behaviors, responses, touch, and movement. These interaction guidelines are related to the Surface design principles covered in the previous section, and they move beyond the boundaries of those principles to include an encompassing perspective on creating Surface applications that are intuitive, engaging, and seamless.

There is a great deal of interconnectedness among the principles; but how much emphasis you place on any given guideline depends on the type of application you're creating.

1. Create experiences for several people to use at the same time

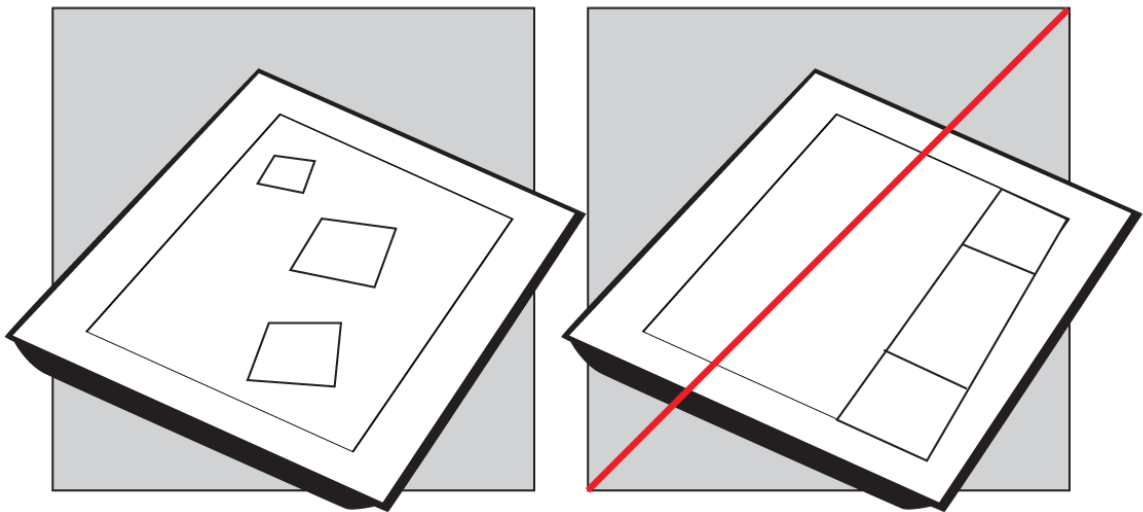
Microsoft Surface recognizes and responds to over 50 different touches at the same time. It sees fingers and objects touching the screen. This enables several people to gather around Surface and share applications, elevating solitary activities to social experiences.



a. Use 360° degree application design for horizontal deployments

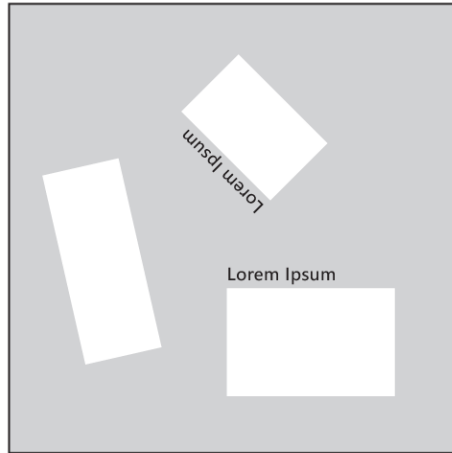
Surface recognizes touch orientation – it sees which direction fingers and special objects are pointed as they contact the screen. This enables developers to generally determine which side of the screen a particular person is on. People will use Surface from all sides so it's important that horizontal Surface deployments be designed for 360° usage.

- Make sure everyone can use content, avoid facing it toward only one side of the display.

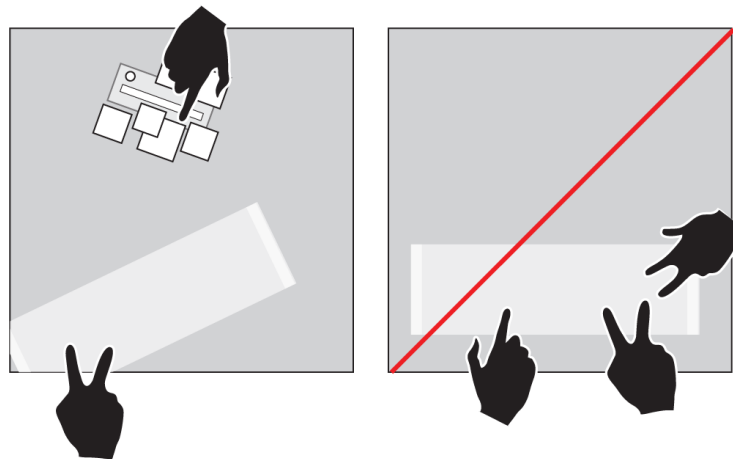


- Enable people to change the direction content faces. Don't lock content to a particular screen direction.

- Use the ScatterView control provided in the Microsoft Surface 2.0 SDK to quickly enable 360° degree experiences.



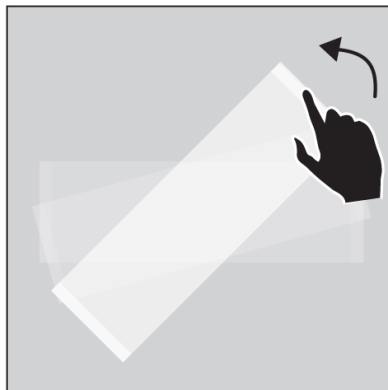
- Make sure everyone can reach and use key application elements from all sides of the unit.



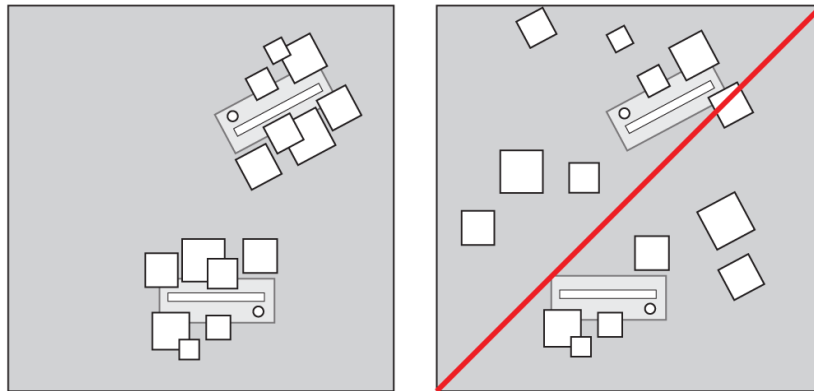
- Some customer application proposals may require people to be on specific sides of the screen. This generally limits application usability, so consider it carefully.



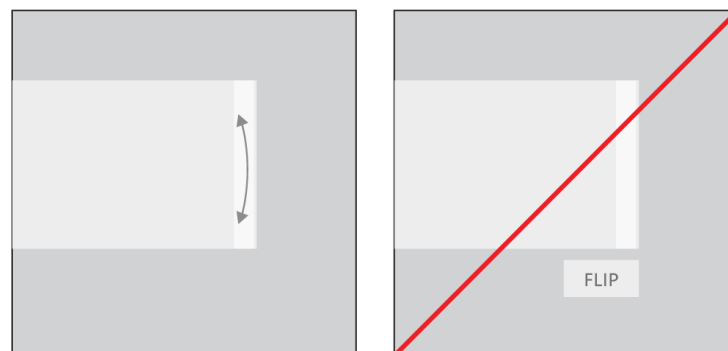
- For example, a collaborative sales application may require a salesperson to always be on one side of the unit and the customers to always be on the opposite side. While this may seem to meet the initial customer need, it will diminish the application as it limits its social appeal.
- Ensure the interface is truly accessible from all sides of the Surface unit. Start using and testing the application in various orientations from the start of application development. Make sure many people share the application at the same time throughout the testing process.
- If the application is freely oriented (has no top or bottom), enable people to change content orientation within the application.



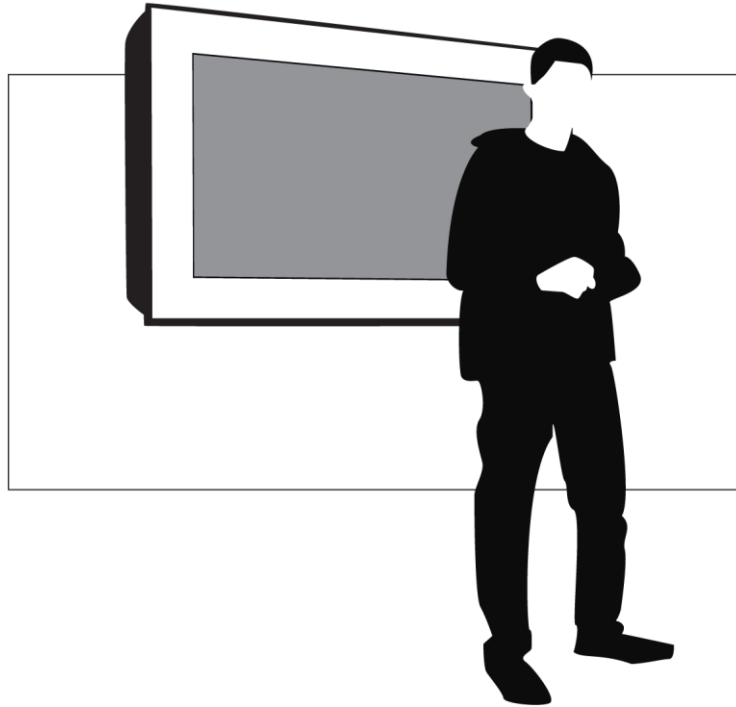
- Surface applications must face new content and interface elements to the people using it. Developers can use finger orientation and the direction of activated content and controls to make reasonable assumptions about which way to point new content and elements.
- If the application still has a distinct top and bottom, give people a way within the application to change the orientation of the entire application.



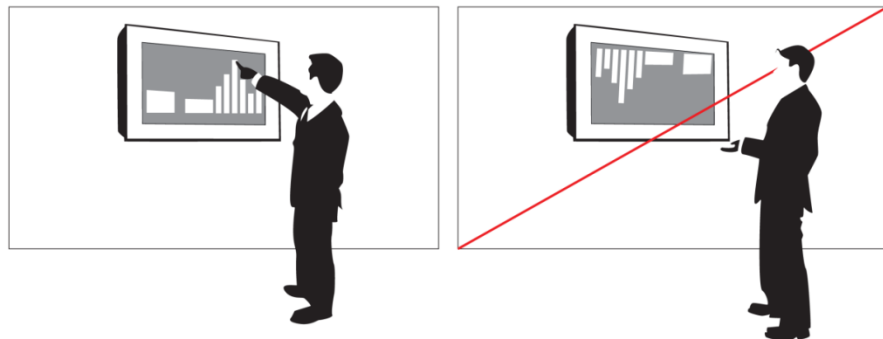
- Simple “flip” buttons are prone to people accidentally touching them, ruining the experience. The Surface Launcher uses a side handle which requiring people to perform a simple, deliberate, action before the grouped content is rotated.



- b. Use 180° degree application design for vertical deployments



- Orient new content and interface elements toward the bottom of the screen.
- Make sure new content appears right-side up.

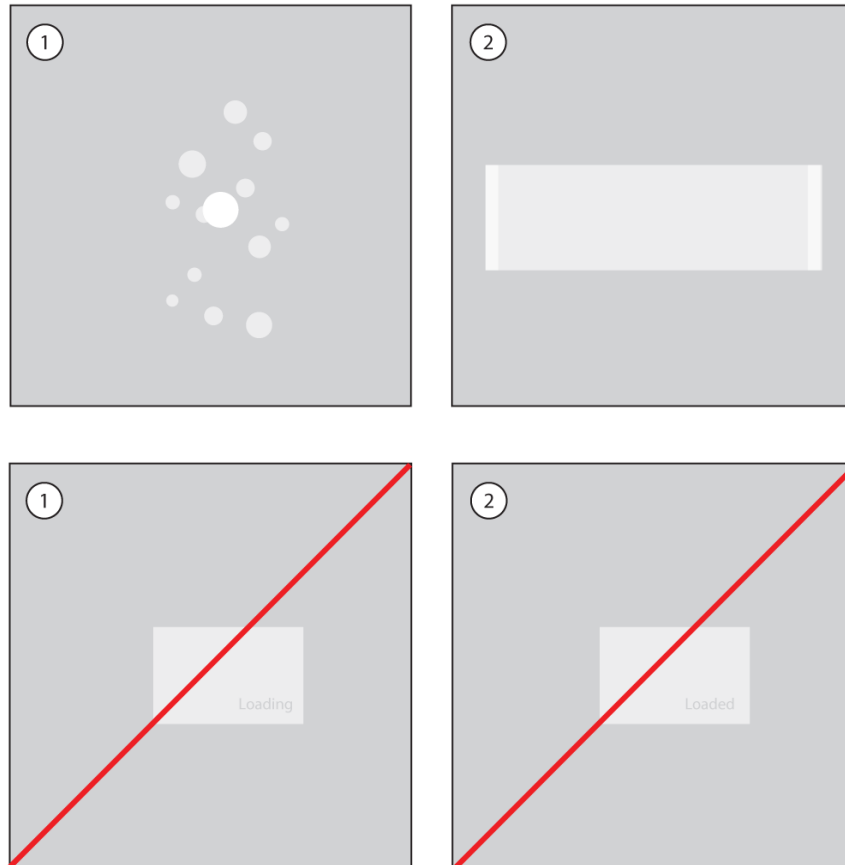


- Understand how people share vertical space.
 - In vertical settings screen space might be sectioned off in columns for several people to use at the same time.

c. Understand how people share screen space

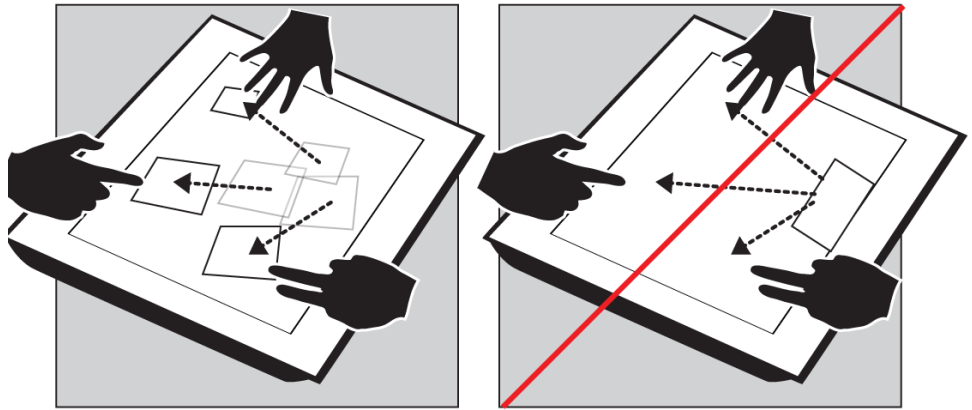
Applications designed for many people to use at the same time need to make it easy for more than one person to see, reach, and touch content.

- Make it clear to everyone when an important application wide change occurs.



- Make sure one person's touch interactions don't unexpectedly affect other people's experience.
 - For example, it's problematic in a paint application if one person selects an eraser control and causes everyone's touches to unexpectedly start erasing content.

- Enable people to move and share controls; do not attach shared controls to one side of the display.



- Communicate content possession through placement; if new content belongs to a particular person, place it in front of them on the screen. If the group shares a piece of content, place it in the center.



d. Support different levels of collaboration

There are three distinct levels of collaboration:

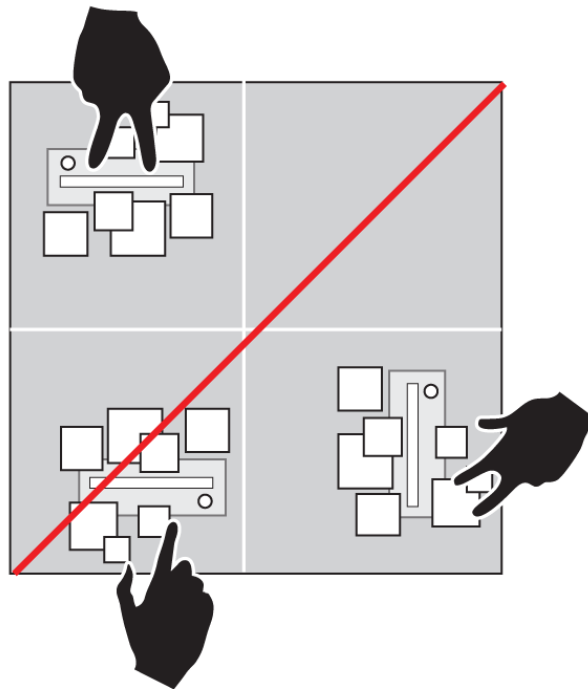
Highly collaborative: People help each other with the same task to accomplish a shared goal.

Somewhat collaborative: People work on separate tasks in order to accomplish a shared goal; also known as "divide and conquer."

Non-collaborative: People engage in separate tasks to accomplish different goals.

Understand how to best support different levels of collaboration within your application.

- Enable several people to simultaneously use content and controls.
 - Do not block progress by requiring everyone to use a common set of controls; allow people to break up portions of the task by physically dividing up and sharing the controls.
- Do not segment the screen into areas tied to particular functions.

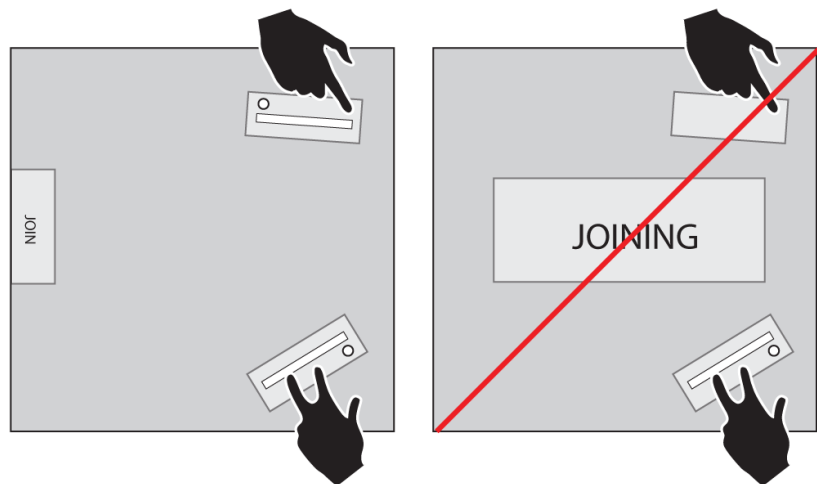


- It is problematic when applications designate one side of the display to perform a specific task, and the other side of the screen to perform another.

e. Understand how people share Surface applications

Surface is optimized for many people to use at the same time, but applications should also be designed for use by a single person. Individuals can use applications and encourage others to join them to share the experience.

- Ensure the number of people using an application doesn't affect its functionality.
 - Enable a single person to enjoy the experience without requiring other people to participate.
 - Enable new people to join without disrupting the experience for people already using the application.



- Make sure your application can continue with fewer people, allowing a person to leave without disrupting all others' experience.
- Create experiences that come alive with several people so that the experience is more fun or efficient when people share simultaneously.
- If appropriate, your application might enable people to divide up their tasks and decide for themselves whether they will be engaged in a collaborative experience or just share the screen to achieve separate goals.

2. Use direct touch interactions

In contrast to the graphical user interface (GUI) mouse and keyboard type of interaction, Surface uses touch and gesture interactions to manipulate onscreen content. Simple

direct touch interactions helps people immediately and intuitively understand the result of their actions, better connecting them to the experience. Using hardware or software buttons to indirectly affect content doesn't typically give them the same sense of control.

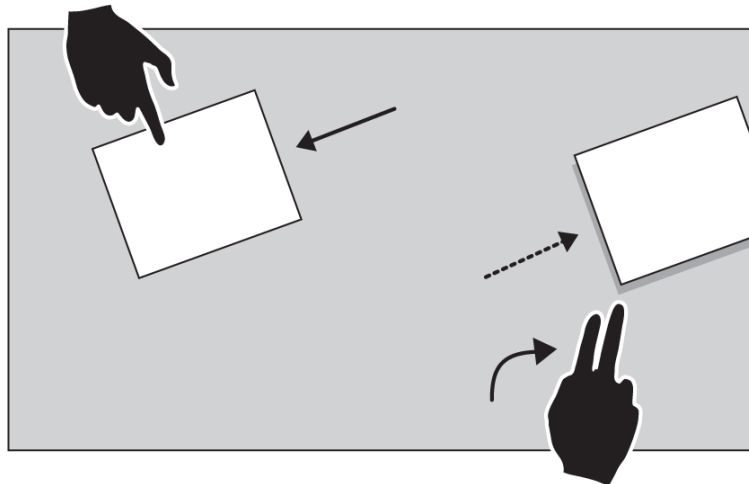
a. Direct touch interactions and indirect touch interactions

Direct touch interactions are physical movements of virtual content within the application by a finger or physical object. Indirect touch interactions usually rely on application interface chrome or abstract gestures. Examples of indirect touch interactions can include buttons, sliders, menus, and gesturing with symbol drawing. Direct touch interactions are the preferred type of interaction for use in Surface because they help to create more intuitive, content oriented experiences.

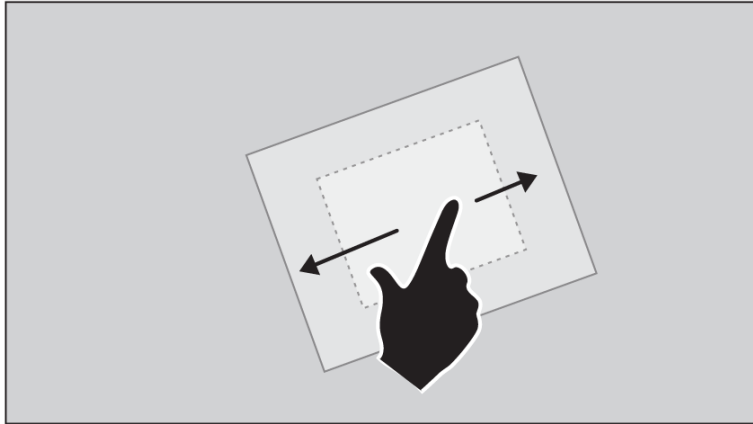
- Do not redefine standard direct touch interactions.

Use the manipulation processor in the Surface SDK to always yield the correct results.

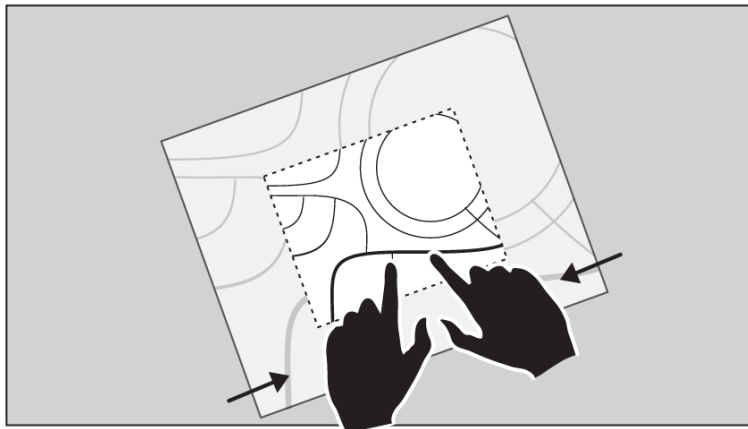
- Move: One or more fingers on an item to move or flick it.



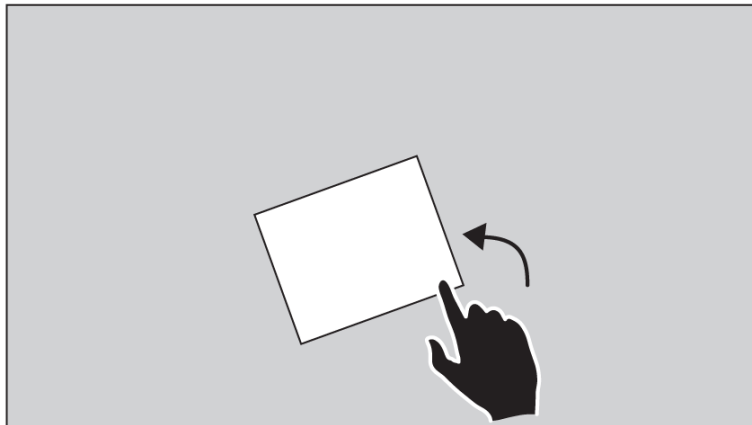
- Resize out or enlarge: Two or more fingers on an item are dragged apart.



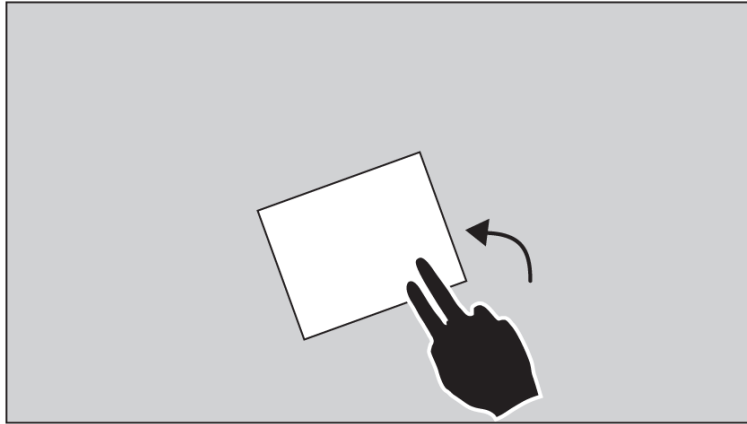
- Resize in or reduce: Two or more fingers on an item are dragged closer together.



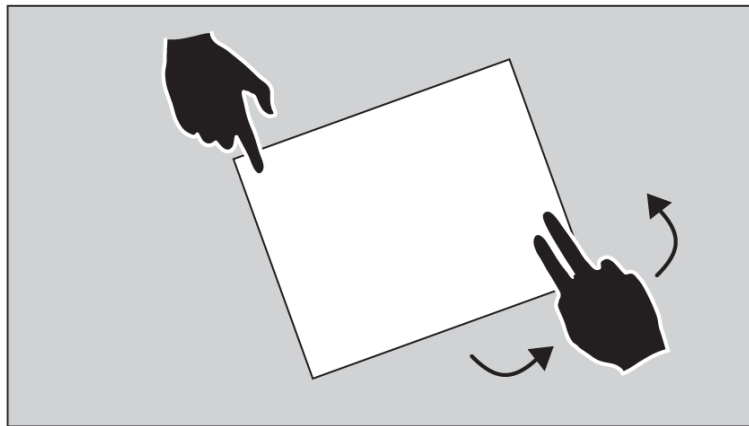
- Single finger rotate: One finger touches an item and drags it around in a circle so that it rotates.



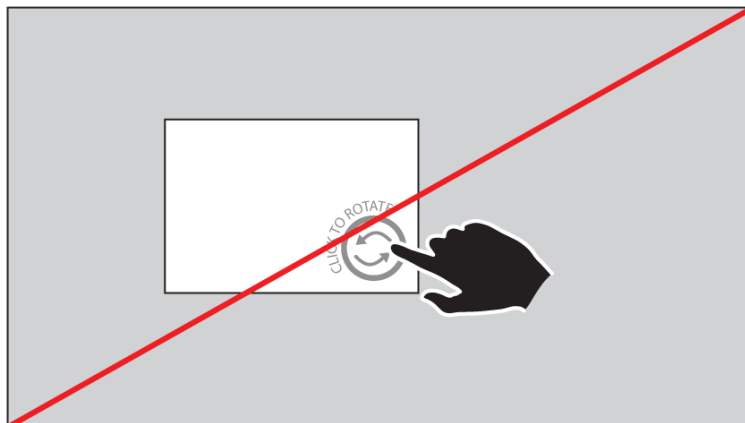
- Two-finger rotate: Two or more fingers on an item are dragged in opposite directions along an arc.



- Pin turn: One finger remains stationary acting as a pivot point while other fingers move around it.



- Don't use chrome or controls to replace direct touch interactions.
 - For example, do not use a rotate button in place of directly rotating an item with a finger.

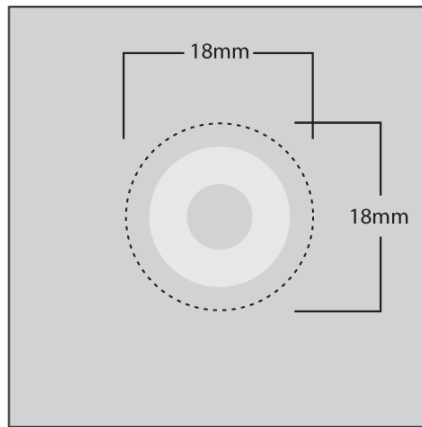


b. Enable single finger drag and flick

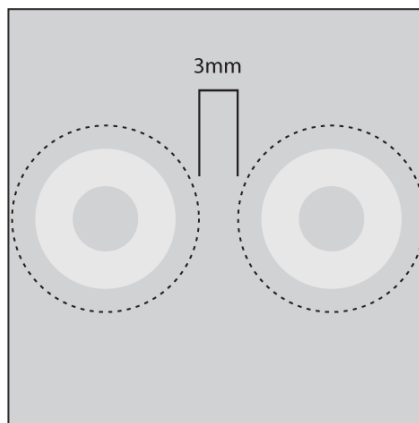
- Enable single finger drag and flick interactions on all movable content.
- Using a consistent single finger drag and flick makes certain that people can always use basic direct touch interactions with all content.

c. Design applications for touch

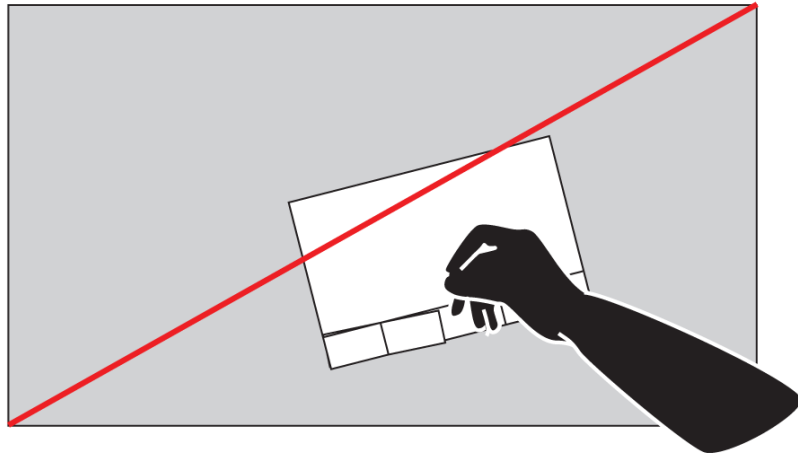
- Interactive elements must be properly sized for finger and object touches.
 - Any item that responds to people's touch, a touch target, must be at least 18mm in size in all directions (18mm x 18mm).



- Allow at least 3mm between minimally sized touch targets.



- Position interactive elements so that hands, arms, and objects covering the screen do not block relevant content surrounding interactive elements.



d. Respond to multitouch

- Microsoft Surface recognizes and responds to over 50 different touches at the same time.
- Your application must respond to multitouch hardware capabilities.
- People expect to use multitouch interactions throughout the experience, not only in self-contained portions of an application.

3. Use physical objects to enhance the experience

People will immerse themselves in Surface experiences, ensure this remains uninterrupted by merging the physical and virtual worlds.

a. Using tagged objects

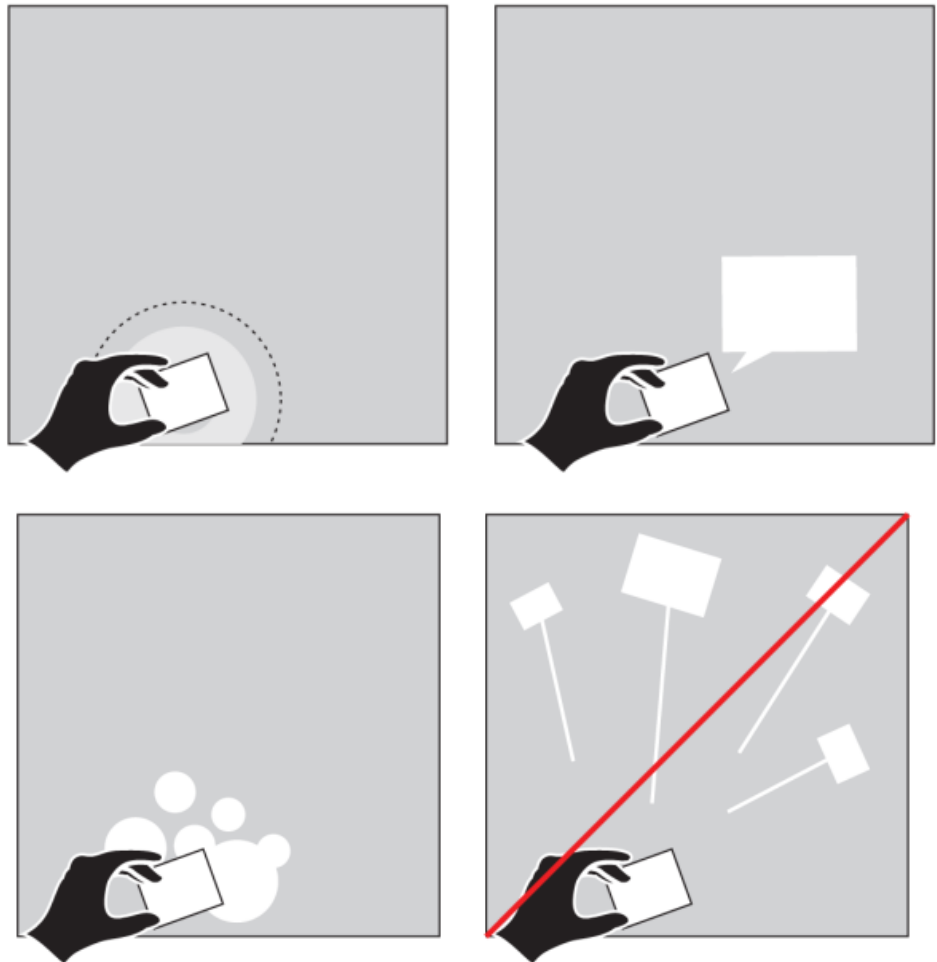
A key differentiator for Surface is its ability to recognize physical objects. Objects help to connect people to the application experience in a tangible way that is not possible with traditional user interfaces.

Placing special tags on an object can help the system quickly and correctly identify it. A tag is a unique pattern of dots that consists of a geometric arrangement of infrared reflective and absorbing areas. When a tagged object is placed on the screen, the vision system reads the tag and determines its value, location, and orientation.

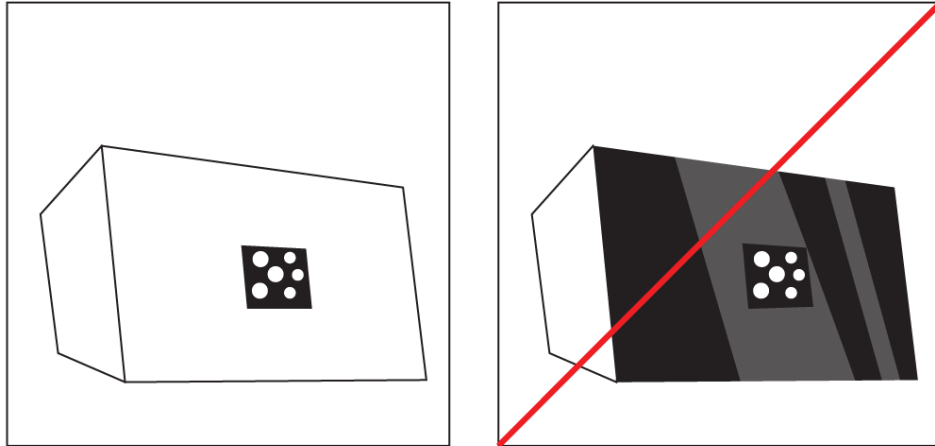
- Use objects to enhance the way people share and own space.
 - For example, an education application might use objects assigned to each student to track lesson progress.
- Use objects to enable playful, delightful experiences.
 - For example, game pieces may have various animations assigned to different pieces. When a player places the game piece on the screen, the tag triggers an animation to play.
- Use objects to create a physical link between people and their content.

Applications that utilize tagged objects should:

- Respond immediately to tagged objects. This immediate response lets people know that the device is working.
- Create a visual response that is appropriate for the object.
- Clearly connect the physical objects with their virtual effects.



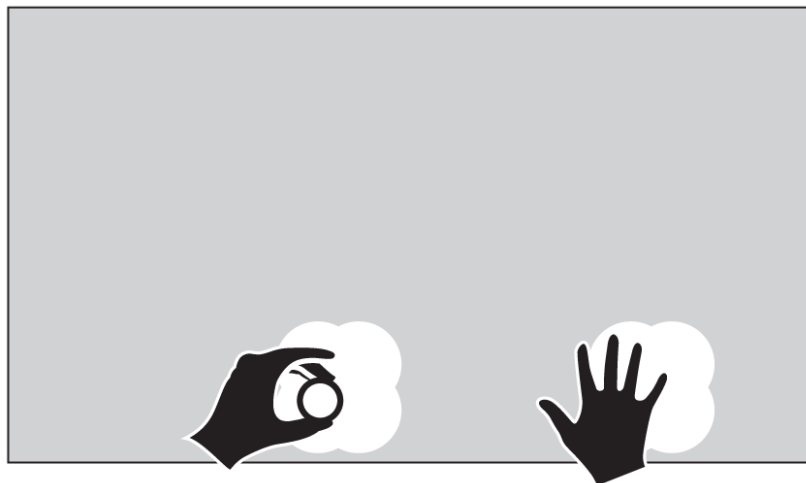
- Use objects to enhance and enrich the experience, but don't require the presence of tagged objects.
- Use non-infrared reflective objects for tagging, so that the reflective portions of the object do not generate contacts within your application.



- If infrared-reflective objects are required, use your knowledge of the tag location and shape of the object to filter (ignore) these contacts.

b. Use untagged objects

An untagged object is referred to as a blob. Surface can detect IR-reflective objects that are placed on the screen. Contacts from untagged objects are the same as contacts from other objects that register as blobs, such as an entire hand placed down on the Surface screen.



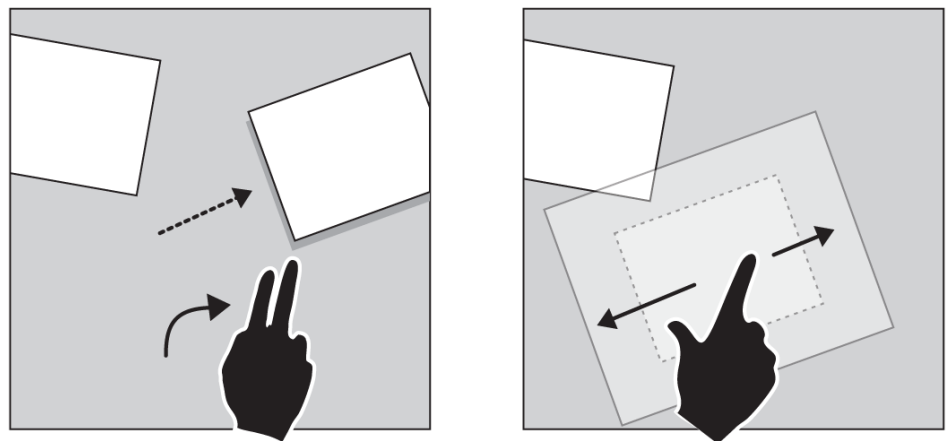
Applications that utilize untagged objects should:

- Respond immediately to the presence of untagged objects. This immediate response lets people know that the device is working.
- Always give a visual indicator when an object is placed on the screen showing that Surface can see it, even if it cannot specifically identify it.
- Do not rely on blob properties to determine the precise shape or size of physical objects. The shape of all objects on Surface is elliptical, and the size denotes only the IR-reflective portions of objects.

c. Make onscreen objects behave like physical objects

Onscreen objects should have subtle physics in performing motions, inertia of movement, and natural feeling collisions to help create a sense of consistency.

- Use interaction metaphors that start with physical manipulation then extend it beyond what is possible in the real world embracing the possibilities of the digital world.



- For example, in a photos application you may be able to move photos around the screen with a flick (mimicking real world interactions) but you can zoom into the photo with a simple manipulation (only possible in the virtual world).
- Mimic the real world in your transitions by using notions such as mass, acceleration, friction, viscosity, and gravity.
 - The ScatterView control in the SDK enables you to easily create these effects.

- Use inertia on objects and content.

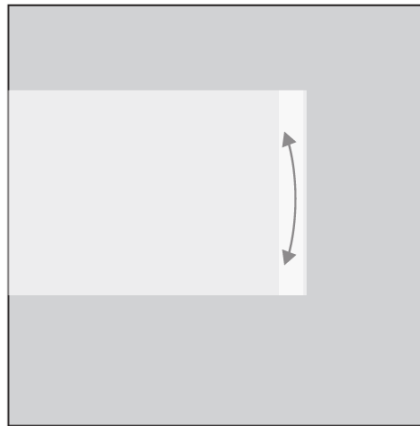
This contributes to creating a consistent experience across applications. The inertia processor in the Surface SDK makes inertia simple to implement. The processor includes two types of inertia: realistic and goal-oriented.

4. Always respond to touch

People might try something that does not work, and the resulting visual feedback should help them learn, resolve problems or encourage them in a positive direction.

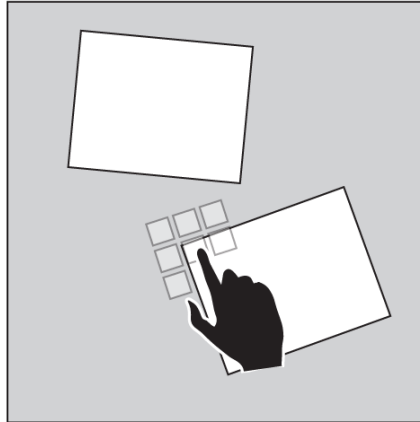
a. Guide people with visual hints and just-in-time chrome

- Give people subtle visual hints as to what the interaction should be.
 - For example, the Launcher rotate handles use a subtle visual hint showing the type of touch interaction that should be used to rotate the control.



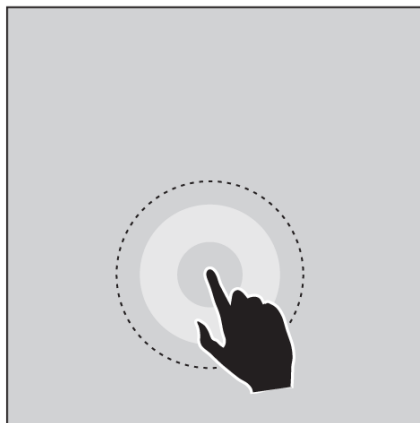
- Use just-in-time chrome to guide interaction. Just-in-time chrome means that controls reveal themselves only when needed.

- For example, in a photos application, ElementMenu should appear upon touching an individual photo rather than being onscreen at all times.



b. Ensure instant gratification and a sense of success

- Visually acknowledge touch.
Touch visualizations quickly and successfully allow your application to visually acknowledge touch.



- All content must respond to touch and must do so immediately and visually. People need feedback that they have successfully touched an object.
- Touch can be acknowledged with depth, glows, or other visualizations.
- Applications must immediately respond to touch input.

- Let people control the experience and do not provide too many automated actions.

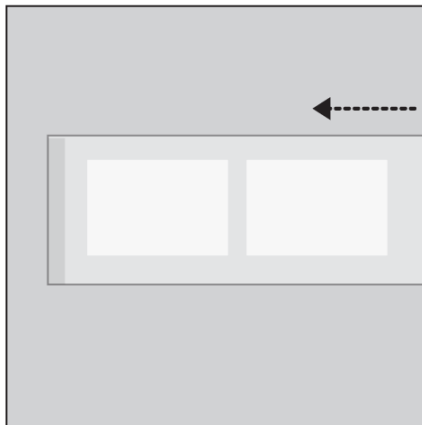
c. Provide engaging sensory feedback

Give feedback that engages people's senses through visuals, sound, motion, and physical interactions.

- Include visuals that are pleasing and appropriate for the scenario.
- Include sounds that are appropriate to the application that help people understand that something has happened.
- Include visual cues and hints that lead people to discover new places, things, or effects within the application.
 - For example, an application can use animation to teach interactions or hint at functionality.

d. Use visual hints and guidance animations to teach touch interactions

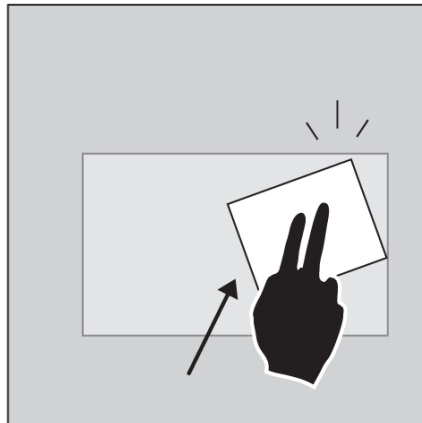
- Provide visual hints for unseen content.



Intuitive, non-menu driven exploration is fundamental to the Surface experience, and sometimes people require clues and hints to guide them in exploration.

- For example, the Launcher animation shows application icons scrolling in and then the icons compress to indicate that the Launcher is scrollable and that there is additional content.

- Provide visual hints that prevent people from doing “wrong” actions.



- For example, do not allow people to move content outside the bounds of a display where they cannot retrieve it.
- Use guidance animations to reveal functionality.
 - For example, in a music browsing application, album covers are ScatterView Items, so that people can touch them and flip them over to reveal the contents of the album. When the application is launched, the covers animate into place and a few albums flip over to demonstrate that additional content exists.

5. Create experiences that are built on content

Content oriented applications encourage exploration and discovery. Experiences that are content focused are more direct, intuitive and enjoyable for people to use.

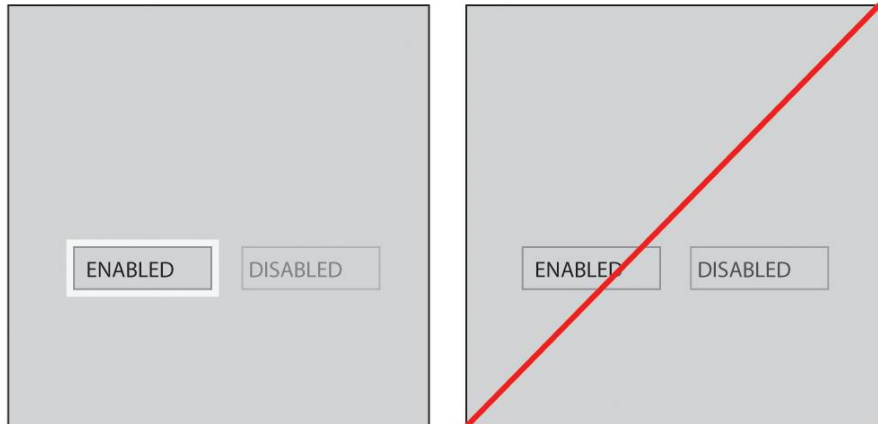
a. Make content the interface

- Do not replace direct touch interactions with UI controls such as buttons, menus and sliders.
- Do not clutter the screen with unnecessary and redundant UI controls and chrome.
- Do not replace touch as the primary interaction.
 - Do not connect external input devices to a Surface unit in such a way that they replace the primary interaction experience.

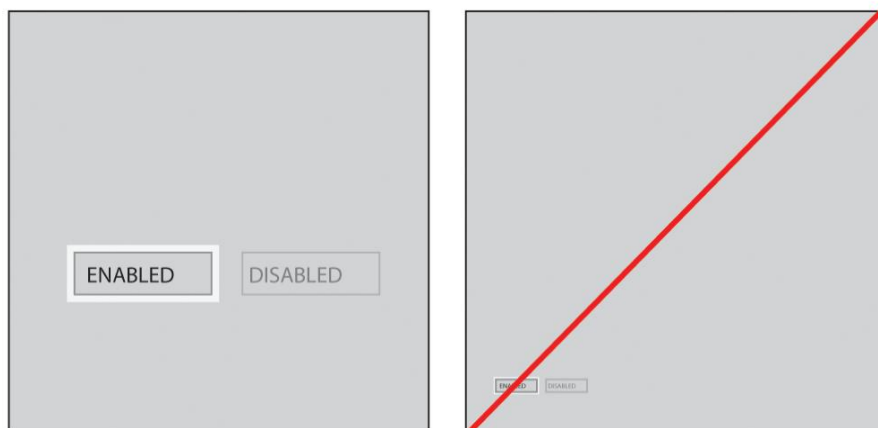
- Applications must support and promote the Surface touch experience, not replace it.

b. Make interactive elements easy to identify

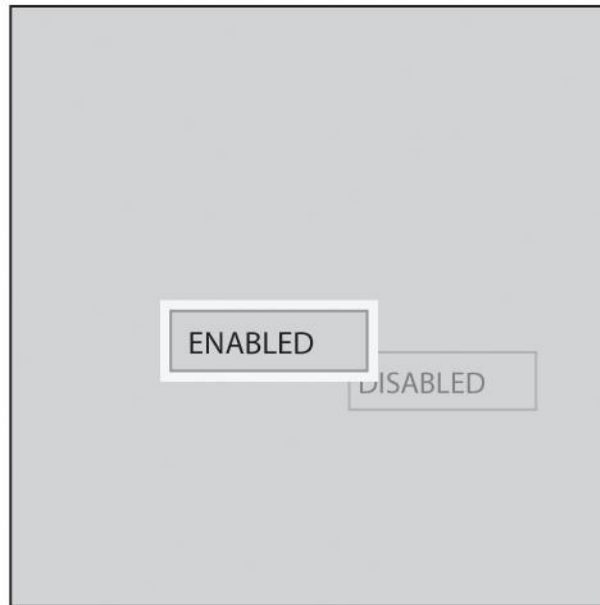
- Make sure visual indicators of touch content are accurate so that people are not misled to touch something that isn't interactive



- For example, a disabled button must be made visually distinct from enabled buttons.
- The larger the interface elements, the more quickly a person can identify it as interactive.



- Use depth to help indicate whether or not an object is interactive.



c. Ensure the experience is focused

- Reduce the number of features in your application so that they account for the needs of the people using the application in a particular scenario.
- Make sure the set of features is focused on particular tasks and that those tasks are clear to people.
- Limit similar choices to reduce complexity and the time needed for people to make decisions.
 - Duplicate controls and interface elements that appear to offer the same functionality are disruptive.
 - When you need to provide a large number of choices, use the ElementMenu.
 - Simple, organized, and hierarchical structure.
 - People can tap or slide to navigate through the menu system.

d. Integrate learning with doing

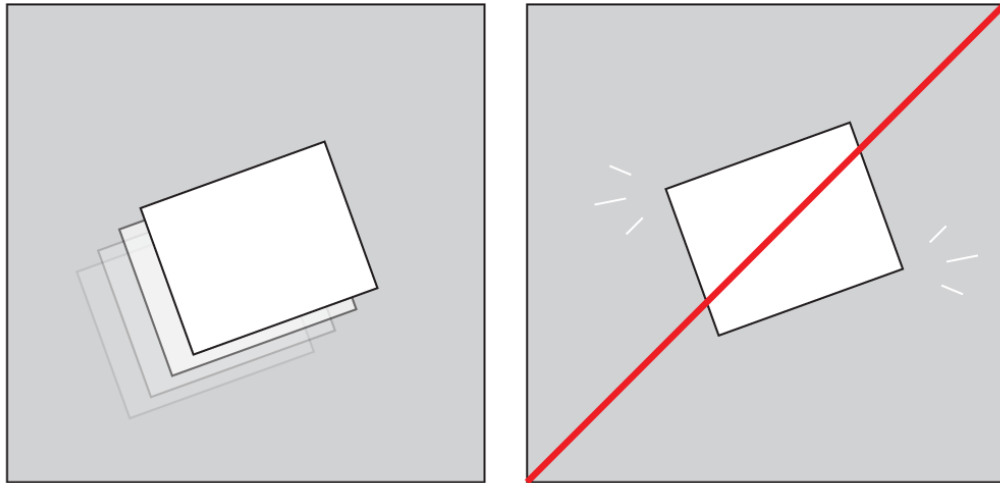
- If interaction instructions are necessary, integrate them with the flow of the application, do not divert focus away from content.

- Make essential features immediately discoverable.

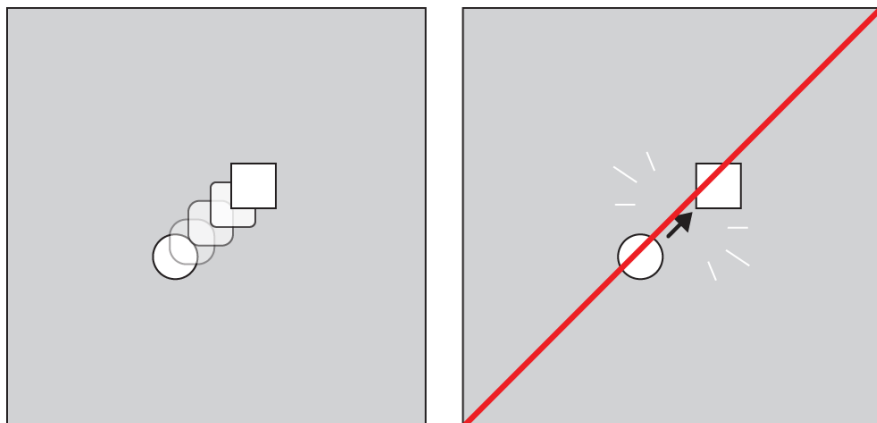
6. A note on transitions

a. Transitions must be fluid and smooth

Smooth transitions give the user context about where they are in the experience.

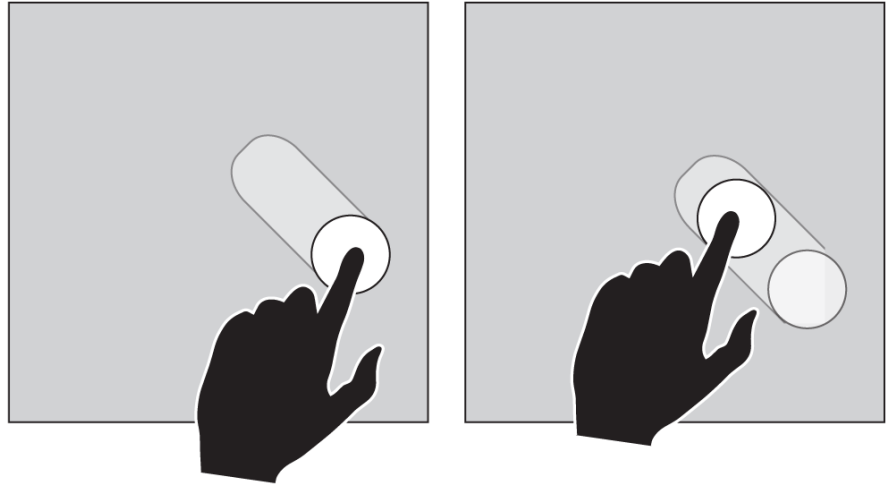


- Make every transition fluid to keep people in a continuous flow.
- Avoid jarring transitions.
 - Smooth transitions maintain continuity and help people stay oriented.
- Nothing should abruptly appear or disappear.
- Every object and visible property change must smoothly animate and transition into and out of existence, or between changes.



b. Avoid unintentional or destructive major transitions

- Require destructive and large changes to be explicitly and intentionally activated by people.



- For example, the Access Points require intentional input to leave an application or start a new experience. This two-step system makes accidental activation of the Access Points unlikely.
- Create a welcoming and friendly application environment that invites people to safely explore and experiment.

Section 3.2:

VISUAL & MOTION DESIGN GUIDELINES

This section provides practical design guidelines regarding visual design and motion design.

1. Orientation & Layout

Laying out visual elements on Surface provides interesting opportunities with the 360° degree nature of the multiuser interface. In most Surface applications, any person from any side of the display should at all times be able to read, understand, and interact with any object on the screen.

There is often no “absolute direction” within Surface applications; there is no explicit top, bottom, left, or right. Instead, it is best to think in terms of position relative to each

person, where an object might simply be rotated or placed towards or away from the people using the device.

a. Application orientation

When an application is launched, ensure that its default orientation is beneficial to most people around the device.

In a *horizontal* deployment:

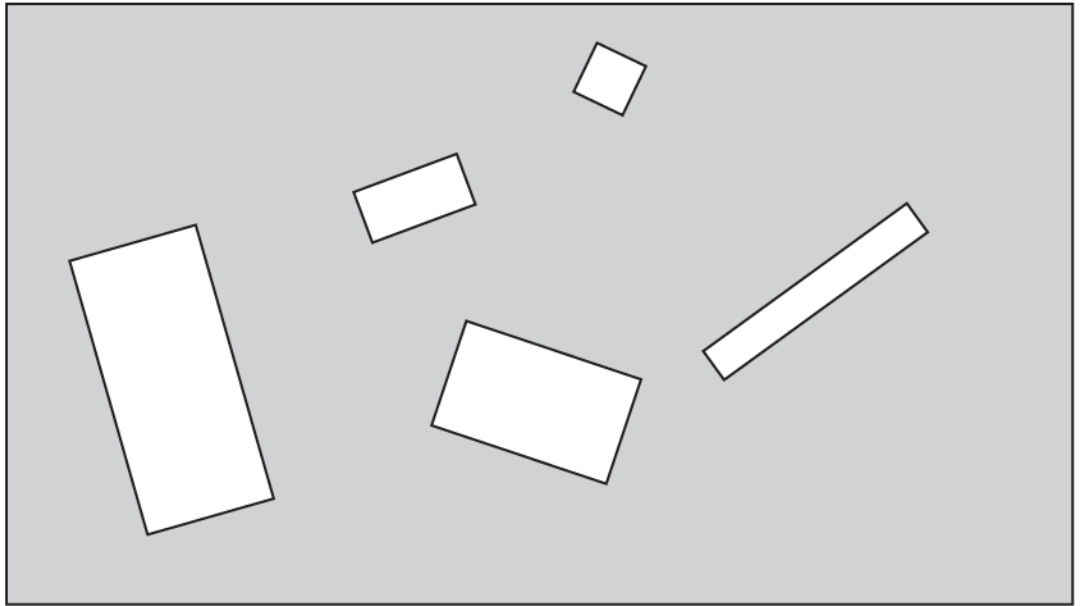
- If an application can have its content freely oriented, that content should be oriented in a way that indicates to people that the content is rotatable and movable.
- If an application must be facing in one particular direction, it should follow the same orientation as the Launcher.

In a *vertical* deployment:

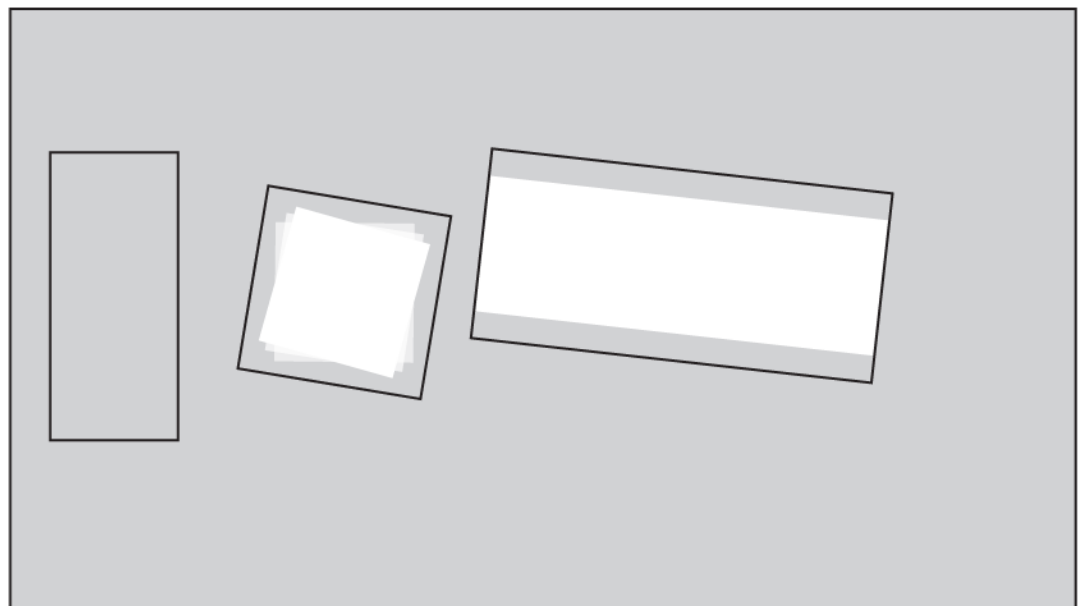
- The Launcher can only be oriented towards the bottom of the screen and applications should follow the same orientation as the Launcher.

b. Grid-free layouts

Most visual designers have learned to create layouts based on grids. The 360° degree nature of Surface is great for laying out applications without a global, or screen wide, grid system. This requires a fresh perspective on visual layout; the Surface SDK ScatterView control is a quick and easy way to create grid-free global layouts. It encourages people to organize and explore content. ScatterView acts as an invisible container for onscreen objects, enabling some content to be oriented towards each edge of the screen by default, which encourages curiosity, direct touch interactions, and exploration.



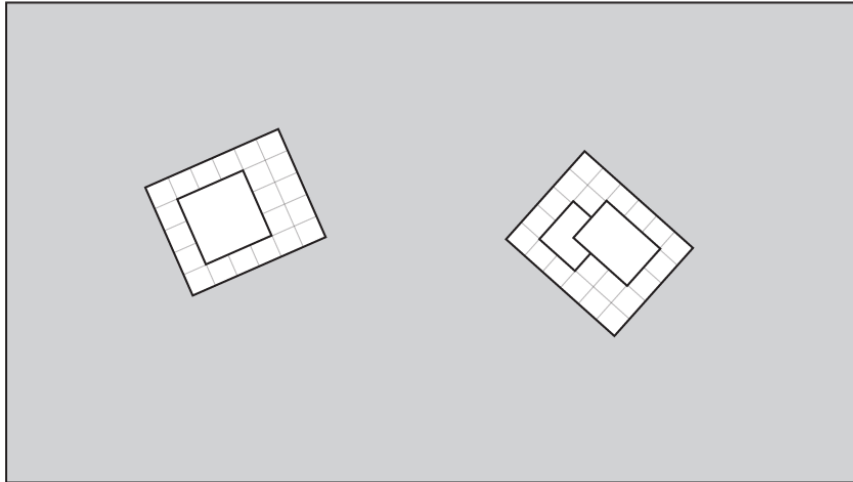
Layouts don't need to be grid-free all the time. For example, the `LibraryContainer` control is a quick way to create order in a local section of a grid-free layout so that content may be easily sorted, filtered, and organized. Applications can also let people decide when to switch between different visual organization methods.



c. Gridded layouts

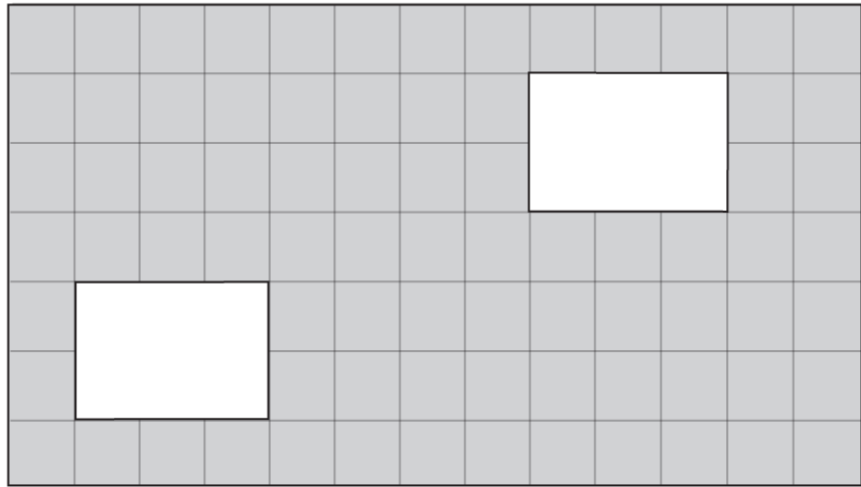
Grid based layouts still have their place and can be useful in certain Surface applications. Gridded layouts are ideal for productivity or focused activities, linear sorting of data, or simply to create a visual rhythm. Layout grids for Surface experiences can be local (specific to a particular content item) or global (screen-wide).

- Using local grids



These are layout systems that apply to specific objects, not to the screen as a whole. A particularly useful implementation of local grids is within a ScatterView item; your application can enable 360° degree experiences with ScatterView, but the local grid can be used within ScatterView to give a feeling of organization and structure.

- Using global grids



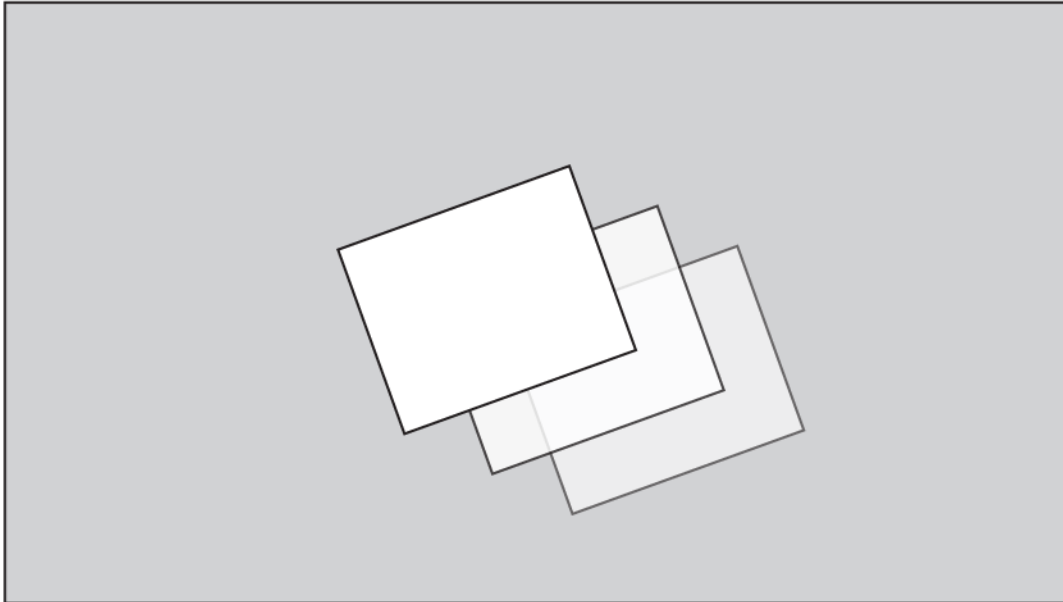
Global grids determine the arrangement of content and controls for the entire application. Screen wide global grids are useful when duplicate controls are offered to each person, or to each side of the Surface screen.

While permitting easy scanning and organization, they often force the entire UI to be oriented towards one side of the screen and make it difficult for people to collaborate. If you are laying out elements on a global grid, give careful consideration to content orientation for every person on each side.

2. Depth

Using the z-axis, or depth, for content organization and visual prioritization can be very powerful. Depth helps separate foreground elements from the background and can also provide clear separation between interactive elements and non-interactive elements.

The Surface interface uses depth throughout the experience and in transitions between various states such as Attract, Launcher, and applications.

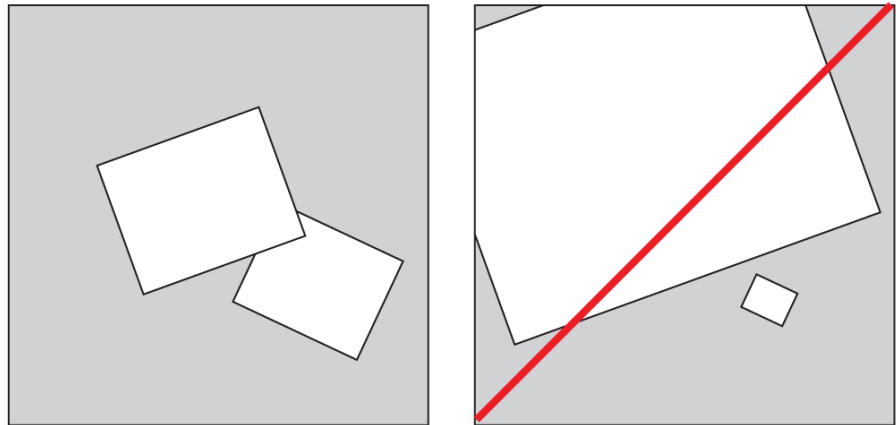


- Use depth to show priority, order, or focus among content items.
- Keep apparent depth shallow, subtle, and elegant.
- Use depth to acknowledge successfully touched objects and controls. Give the illusion of content floating toward a finger to register a successful touch, reserving the highest level of depth for those objects actively being touched.
- Use depth within content and controls to visually indicate touchable regions.

a. Creating depth using 2.5D

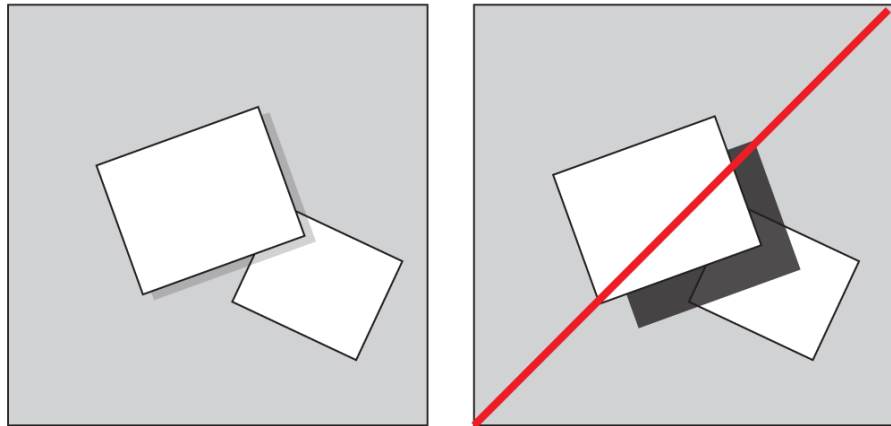
2.5D is typified by using techniques that simulate or create the illusion of depth without actually requiring 3D geometries. The various techniques can be done without specialized 3D tools using XAML and the Windows Presentation Foundation (WPF) in a high performance way.

i. Scale



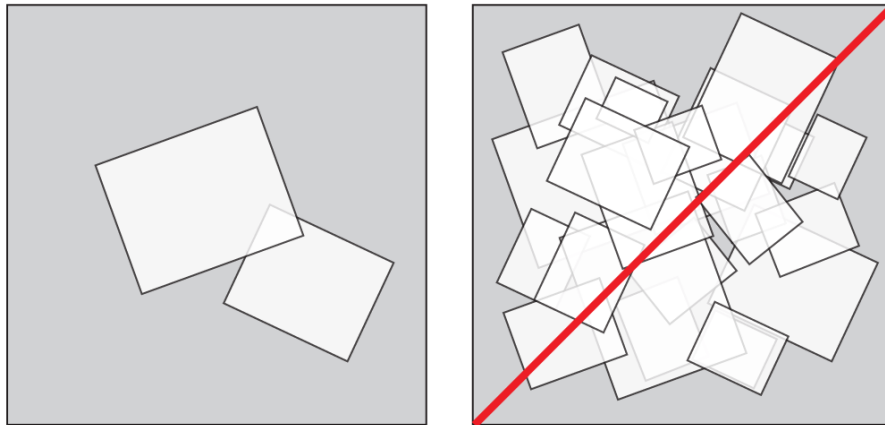
Scale can be used to simulate depth, especially when combined with other techniques. Scaling is an ideal way to register successful touch input; a slight increase in scale (or the illusion thereof) gives the impression of an object being magnetized to someone's finger. Large differences in scale between different levels of depth can lead to an appearance of an extremely deep background. Use small, subtle changes in scale to keep people immersed in the experience.

ii. Drop shadows



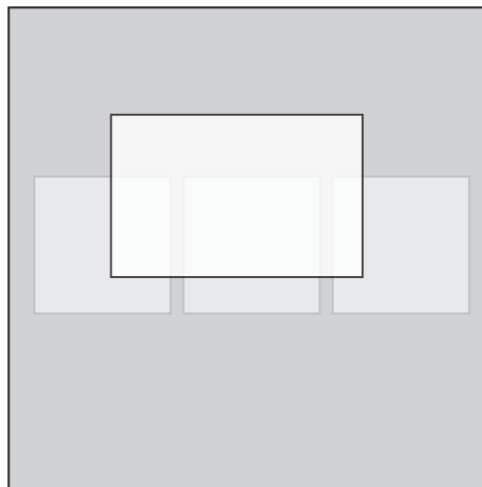
Subtle drop shadows are an easy and effective way to indicate depth, but it is important not to make them too heavy. Dark shadows can obscure content below, and very large offsets can make objects feel disconnected from the Surface display.

iii. Transparency



Transparency can show depth by allowing objects on lower levels to show through objects above them. This technique is good for a small number of objects onscreen at the same time as transparency can be computationally expensive to render.

iv. Depth cueing

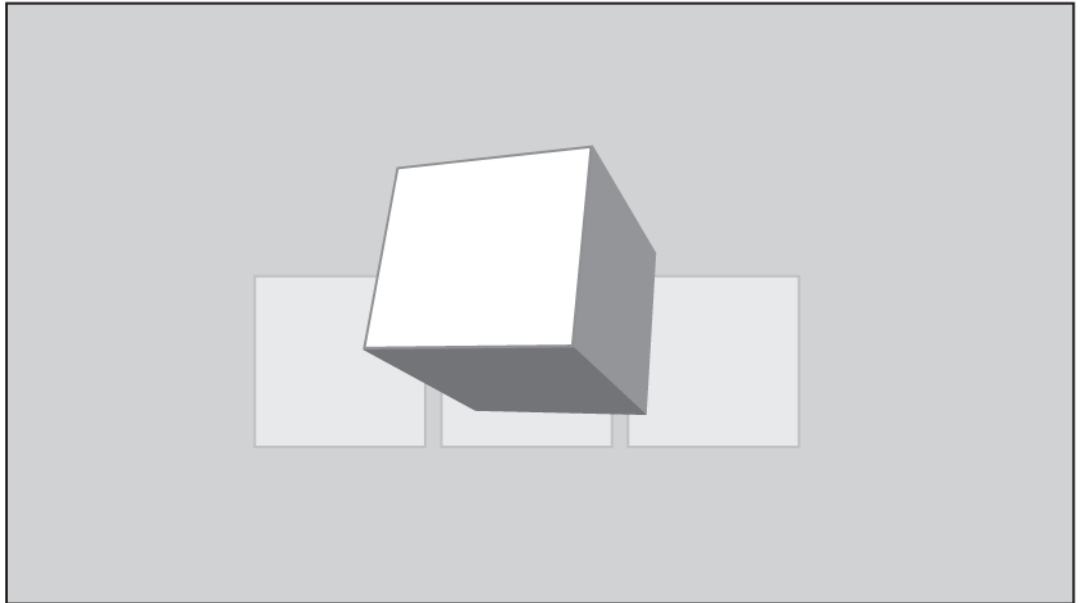


Depth cueing uses tinting, saturation reduction, and/or contrast reduction to simulate distant objects. Subtle reductions in brightness to convey depth and object focus can be quite effective.

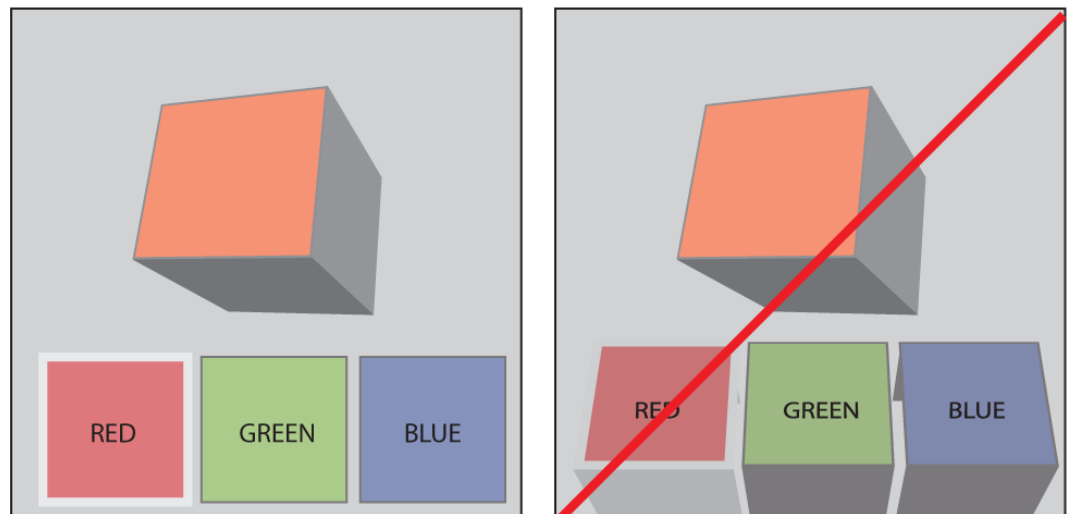
b. Creating depth using 3D

True 3D uses rendered three-dimensional geometries in real time. This enables realistic rotation of cubes, spheres, custom 3D models, and so on. While 3D can

be authored in XAML and delivered using WPF, it can adversely impact application performance. True 3D is best created and delivered on Surface using XNA, the core Microsoft 3D and gaming engine.



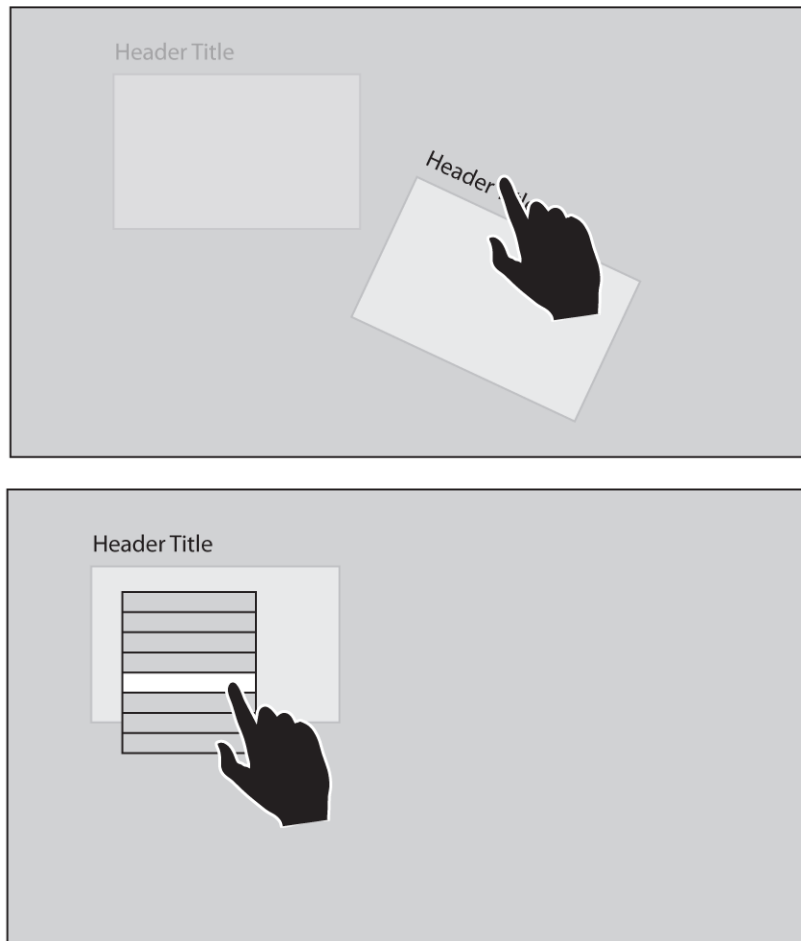
True 3D offers many opportunities for interface innovation, but use caution when designing 3D content. Controlling a 3D object on a two-dimensional touch surface can be difficult and confusing. 3D should usually be avoided for the creation of controls, as any text or labels on 3D surfaces can become difficult to read.



Consider using 3D depth not to describe onscreen objects, but instead as a virtual space to guide people through the application.

c. Depth as visual hint

Depth can also be used to indicate interactive zones for touch input. This helps people understand that touching in one region will have a different result than touching in another. For example, the header or title bar of a content item is likely to be dragged, but a list item within that object will scroll when dragged. Making one portion of a control appear above or beneath other elements within that control can help people identify these zones and make their purpose easier to understand.



3. Shape & Form

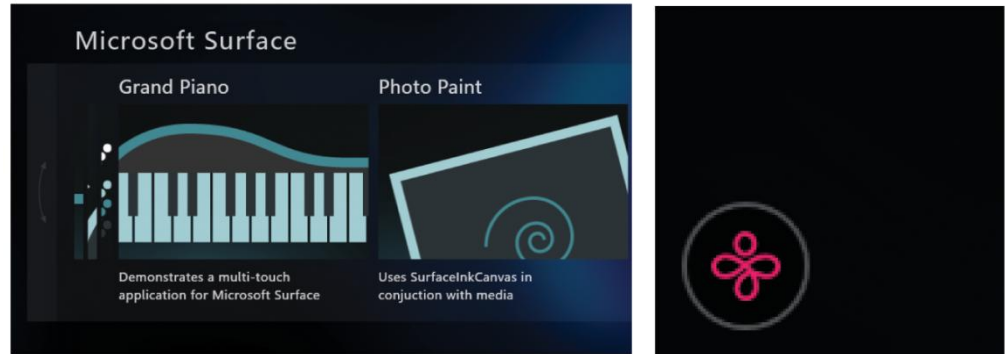
The shape and form of Surface content should visually indicate its function.

a. Edges and corners

The styling of content edges and corners in applications will affect the kind of experience people have.

Looking at the core Surface interface, two styles are used to communicate different messages to people:

Style 1: Flat



The Surface Access Points and Launcher are meant to be clean, flat, simple, and digital. This style indicates that the Access Points and Launcher are interactive elements used to navigate through the Surface experience. These navigation controls are secondary to content. They are not distracting, and they help people get to the next phase of the experience.

Style 2: Organic



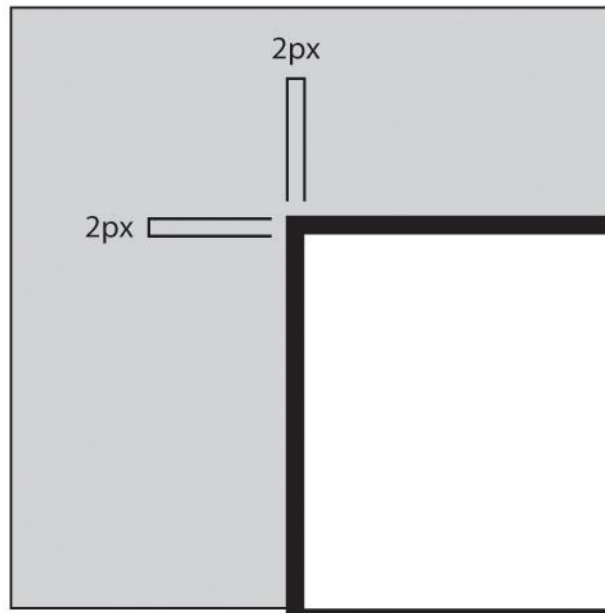
The Surface Attract mode is based on volumetric, organic forms that encourage exploratory touches. Rounded, organic forms are generally preferable for approachability, discoverability, and engagement.

Balancing these styles will help achieve an experience that is appropriate for your application.

b. Line work and outlines

The Surface interface is typically free of outlines, keeping focus on the content.

If an interface element requires outlines, keep line work to a two pixel minimum width to produce best results when rotating or scaling.



4. Texture

Generally, using textures as pure decoration is not recommended for Surface experiences. Ornamental textures tend to detract from content, which should always be at the core of the experience. Textures used for purely aesthetic reasons typically create experiences that are not authentically digital.

There may be certain applications where textures themselves *are* the content. If texture must be used in your application, keep them minimal, consistent, and always stay focused on encouraging touch and exploration.

While textures can act as visual hints to help people use the application, many other visual design techniques can achieve the same goal: negative space, form, shape, color, and more. Explore alternatives to using textures as visual hints.

5. Colors

Designers should follow established color design guidelines, and take into account the specific requirements of the brands they are handling. There are some additional guidelines that optimize color reproduction for the Surface display.

a. Using colors

- Proof all colors on Surface during the design process; iterative proofing is the best way to achieve consistency and optimize brightness and color for the Surface screen. Do not rely on a desktop monitor for proofing.
- Never use pure white as a background color.
- Bright or saturated tones are best in smaller regions to reduce eye fatigue.
- High contrast, such as pure black on pure white, reduces the effectiveness of anti-aliasing (the subtle edge blending of foreground elements onto the background), making text look harsh and introducing a jagged stair-stepping effect into object and content edges when rotated.
- Treat the background as a stage against which all action occurs; it should be subtle, calm, and understated.
- Use subtle gradients cautiously; the reduced color contrast may cause subtle gradients to flatten and appear as one continuous tone.
- Test gradients on the Surface screen to check for banding.
- Experiment, iterate, and explore as you become familiar with the Surface display characteristics. Keep an open mind towards taking fresh approaches to a brand's established color palette.

b. Surface colors

Surface administrators control the default color scheme of the Surface system. This global color scheme determines the default colors of Surface controls that are rendered in all applications. You can use those same colors to style additional elements in your application. However, you can also override the Surface colors to affect any element in your application that uses those colors.

Surface comes with the following four pre-defined color palettes, defined in the classes of the `Microsoft.Surface.Presentation.Palettes` namespace:

- `Light palette`
- `Dark palette`
- `LightHighContrast palette`
- `DarkHighContrast palette`

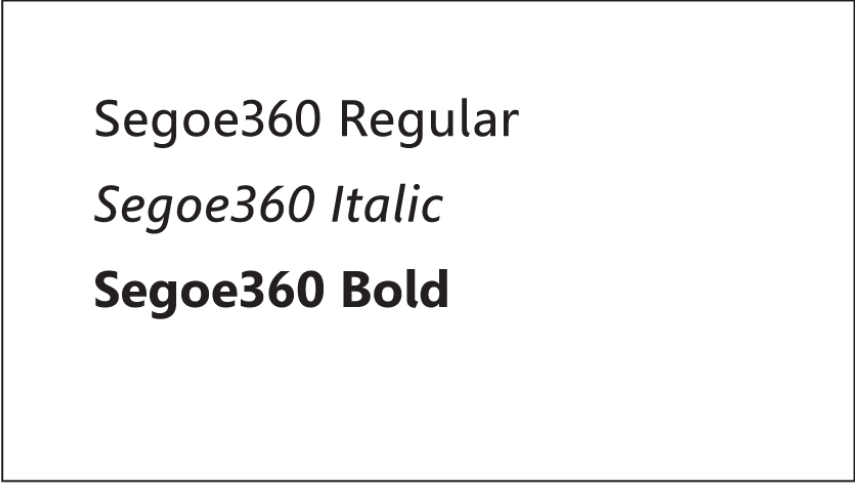
6. Typography

Typography in Surface applications should be elegant and meaningful. With simple interface elements and content at the core of the Surface experience, onscreen text has even more opportunities to act as the interface itself.

a. Segoe360

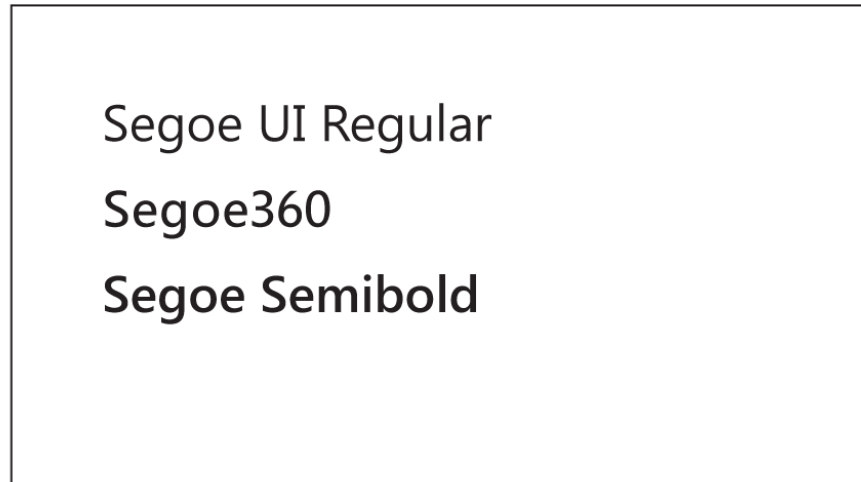
Segoe360 is a specially designed typeface for use on Surface. It's strongly recommended that you use Segoe360 in your applications for the following reasons:

- Segoe360 comes with three weights- Regular, Italic, and Bold.



Segoe360 Regular
Segoe360 Italic
Segoe360 Bold

- Segoe360 Regular is a new weight between Segoe UI Regular and Segoe UI Semibold, designed to have better presence on Surface.



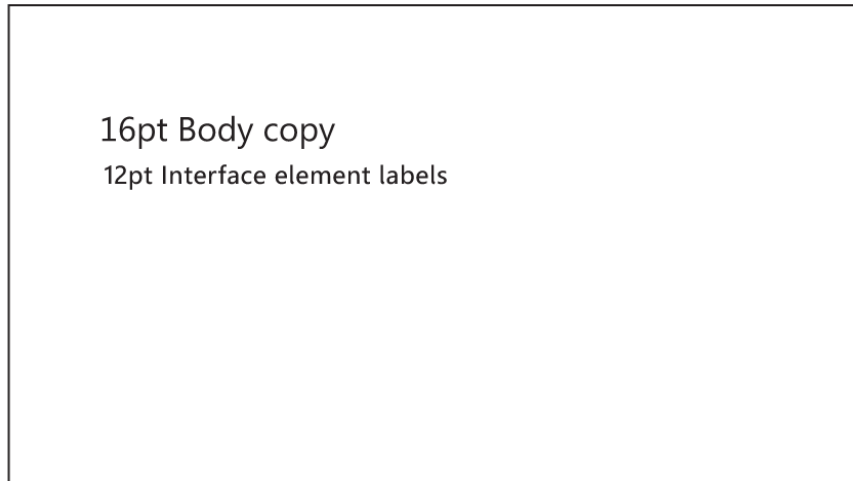
- Characters have been modified for better distinction at all viewing angles, enabling better 360° degree readability. For example, the lowercase “l” is more easily distinguished from a capital “l” and a lowercase “q” is more easily distinguished from a lower case “b” when rotated.



- Tracking (overall letter spacing) has been increased to be more legible and recognizable onscreen, especially at small sizes.
- Counters (the enclosed areas in letters like ‘a’ and ‘e’) have been opened up so they retain their visibility at small sizes.
- Segoe360 supports Western Europe, Eastern Europe, Turkish, Baltic, Greek and Cyrillic character sets.

Minimum Segoe360 type sizes:

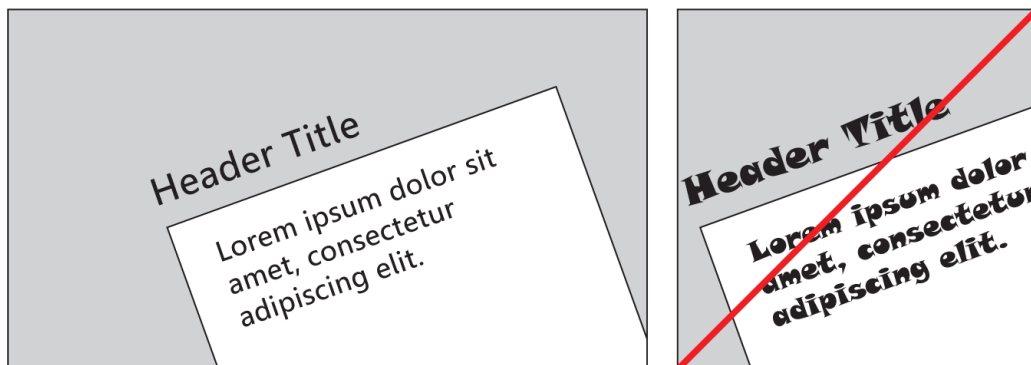
- 12 points for interface element labels
- 16 points for body copy and text



b. Selecting typefaces

Segoe360 is strongly recommended for use in Surface applications, but it may not always fit the needs of a particular brand or scenario.

When choosing an alternative typeface you need to consider rotation, legibility, and orientation. Sans serif typefaces – those without small projecting extensions at the ends of character strokes – are typically preferred over serif typefaces, due to their minimal flourishes, consistent widths, and generous negative space. Take great care when selecting and using alternative typefaces for Surface; be sure to test legibility on a Surface display before finalizing selections.



- Evaluate text rendering at all angles before and during application development.
- Avoid using all capital letters and small capital letters. Sentence case capitalization is more natural, easier to read, and rarely conveys an aggressive tone.
- If serif fonts are an integral part of a visual brand experience, set them at 30 points or larger for maximum legibility.
- Typesetting in high-contrast reduces legibility. White text on a black background can create harsh edges and a poor reading experience.
- Avoid setting text on curved paths.
- Remember that most Surface objects and controls can be scaled freely. Enforcing minimum scaling limits on objects is sometimes necessary to retain maximum text legibility.
- There is no substitute for testing font rendering on Surface itself. Make sure to test for legibility, including when text is rotated and scaled.

Minimum alternative type sizes:

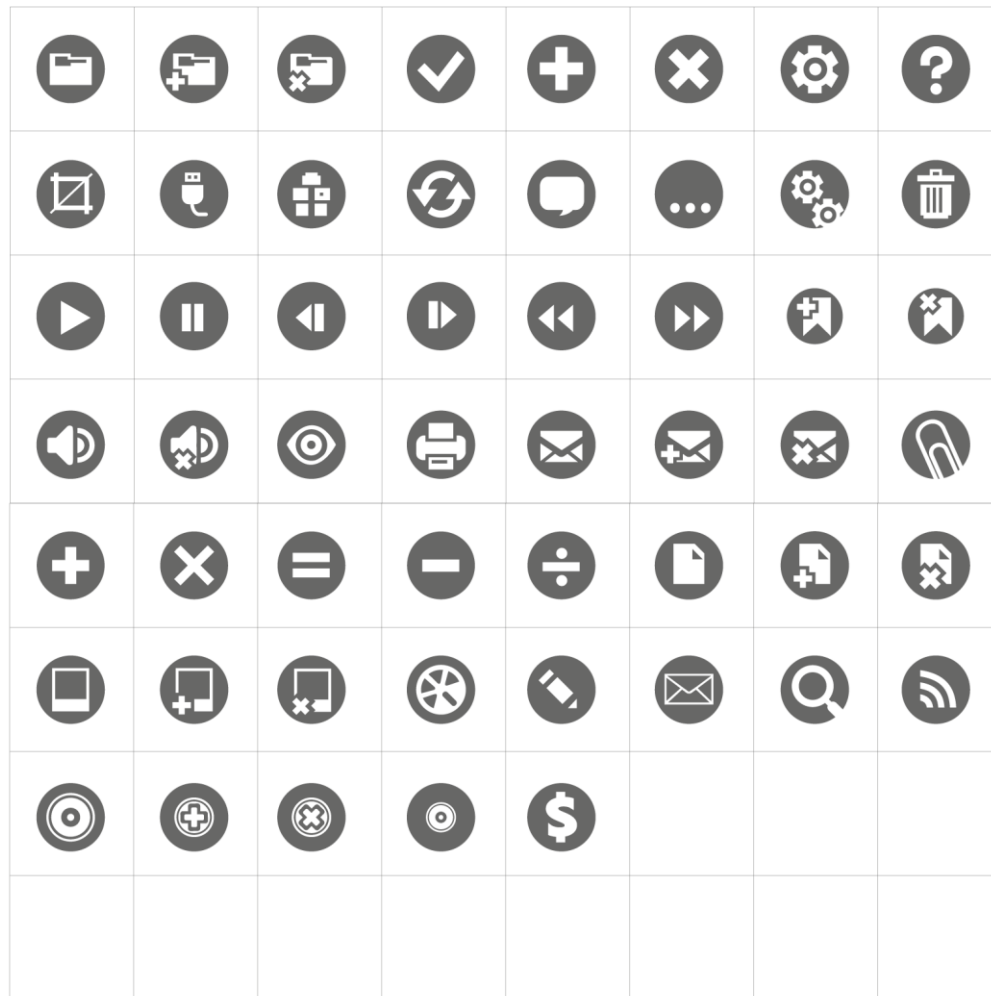
- 12 points for interface element labels
- 16 points for body copy
- 30 points for light typefaces (thin and/or narrow width)
- 30 points for serif typefaces (containing small projecting extensions at the ends of character strokes)

7. Iconography

Icons can help the Surface experience, but usage must be meaningful.

a. Icons as visual hints

Icons are useful to visually convey functionality. The function of buttons and other controls is not always obvious to people, and it is not always feasible to use text labels on small controls. In such cases, simple yet bold icons can be used as a supplement to textual button labels. A set of more than 40 sample icons is distributed with the Microsoft Surface 2.0 SDK.



Do not use icon buttons when suitable direct touch interactions are available, use icons to hint at functionality. For example, the Surface Launcher is rotatable. Rather than placing a button with a rotate icon in the corner of the Launcher, the Launcher uses subtle rotation arrows to indicate the action people should take.

b. Creating custom icons

If an icon is needed for your application that is not included in the sample icon set use the following guidelines to develop custom icons:

- Icons should have simple geometry.
- Icons should have limited detail.
- Icons should be scalable.

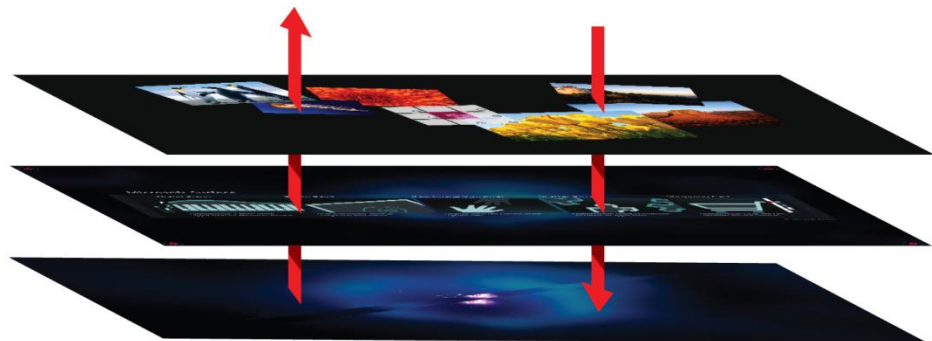
- Icons should leverage real world metaphors.
- Icons should have a similar visual weight.

8. Motion Design

Motion design defines how things move onscreen, a critical part of the Surface experience. Animations always support the content and the experience as a whole. Motion design provides animations and effects that convey emotion, energy, connection, and responsiveness. They provide visual hints, cues, and an invitation to explore content.

a. Transitions

Transitions can give context and provide visual hints of where people are in the Surface experience. They help people build mental maps of their experiences, learn how to use controls, and remember where onscreen objects have gone when no longer visible.



The Surface transition model for shifting between various modes (Attract – Launcher – applications) uses depth to provide navigational context. For example, when you leave Attract to enter Launcher, Attract pushes back in z space – it moves behind Launcher. When you launch an application from the Launcher, the Launcher moves back in z space and pushes Attract back even further. This gives people important context regarding the way they are moving through the experience.

Develop a transition language or model that is appropriate for the function, character, and style of your application. Consider the type of application that you are developing and create transitions that are fitting and appropriate.

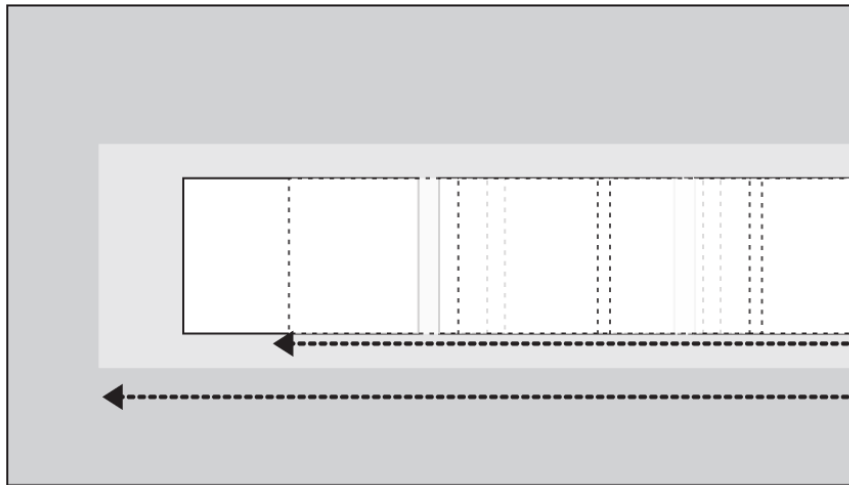
Transitions that occur in a game may be fun and playful, while transitions that occur in a healthcare application should be brief and succinct.

Transitions should:

- Be seamless
- Be meaningful
- Be informative
- Be application appropriate
- Be carefully timed. Don't frustrate people by forcing them to wait through long transitions just to access application content.
- Provide navigational context
- Reinforce the application narrative

b. Transitions as visual hints

Motion design provides tools that enable people to rapidly learn through observation; without requiring directed tutorials or lengthy demonstrations.



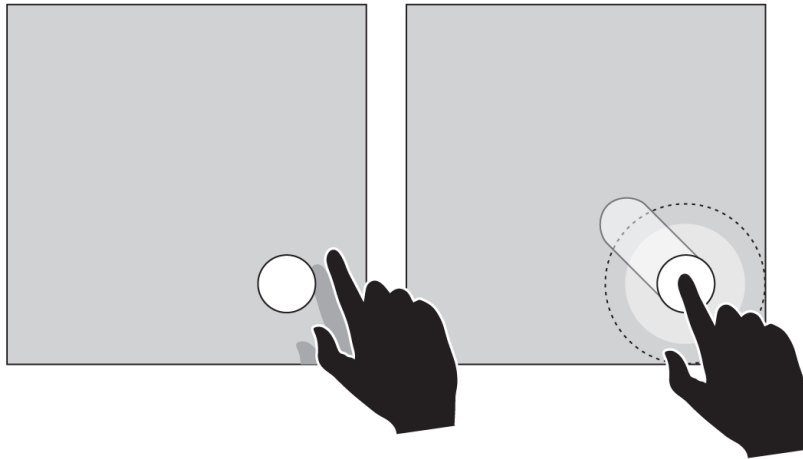
For example, when the Launcher loads a background container fades in, the application icons slide in, and the application icons compress as they reach the edge of the background container. People see this movement and understand that there are more options than are shown. The sliding movement visually hints that the content is scrollable.

If the Launcher container suddenly appeared onscreen without any transition, it would seem deceptively static. It would be hard for people to know that the

content is scrollable, and that there are more options available than those currently displayed. It is very valuable to use transitions as a way to visually hint at functionality.

c. Guidance animations

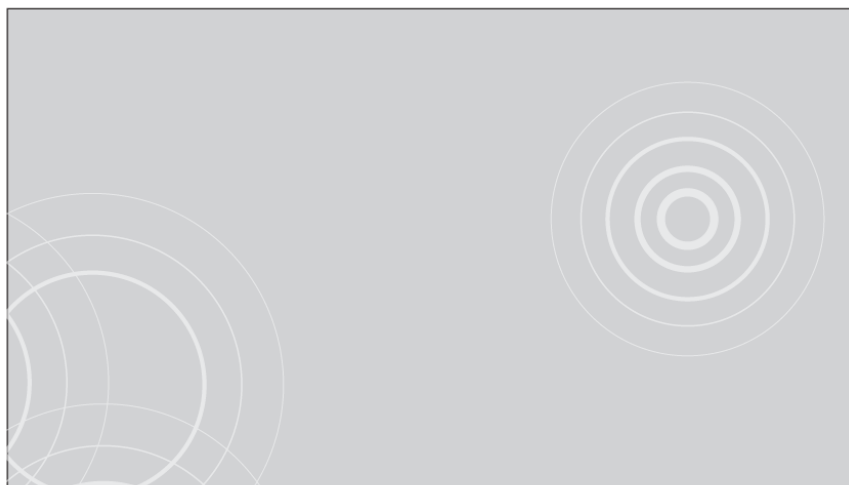
Guidance animations teach people how to perform specific touch interactions to accomplish a task. They communicate the way a particular button icon or piece of content should be used and give people feedback to visually explain a task.



For example, the Access Points use guidance animations to teach people a specific direct touch interaction. When someone touches the static Access Points, a visual indicator and action word animate out of the Access Point. If the person does nothing, the visual indication fades out. If the person slides the Access Point or taps the action word, they navigate to a different part of the Surface experience.

d. Ambient animations

Ambient animations are subtle animations used to indicate that the system is alive, awake, and ready for use. They can help reinforce brand and application personality. Ambient animations are purposefully unassuming and refined using subtle and natural effects.



Subtle ambient movement in the background can give a Surface experience a lively quality. Use ambient animations in your application to ensure people that the application is responsive and working.

Section 3.3:

SOUND DESIGN GUIDELINES

Sound design for Surface should draw heavily from the Surface design principles. Adhere to the following guidelines when developing or using sounds for Surface applications.

1. Make sounds simple and subtle
2. Ensure sounds are authentically digital
3. Use sound judiciously. Sound is often difficult to hear in public locations. If sounds are overbearing or annoying people will mute the speakers or remove the application.
4. Make sure all application sounds are at a level (volume) consistent with the sounds in the Surface Shell and Launcher. If sounds suddenly spike in the middle of an experience people will mute the speakers or remove the application.

LANGUAGE & TEXT DESIGN GUIDELINES

Surface experiences typically require little interface text, but this will vary with the type of application you are developing. Text should be content oriented and used judiciously.

Using the Surface design principles to influence the personality, terminology, capitalization, and punctuation of text will add an extra layer of polish to applications and significantly enhance usability.

1. Personality and disposition

The personality of text in Surface applications should be lively, clear, and consistent. It should not include soulless, jargon filled, incomprehensible computer-like language that prevents people from enjoying themselves – or asks them nonsensical questions in cryptic dialogue boxes. The disposition of text in Surface applications should be friendly, lighthearted, and empathetic. Application text should not be rigid, snobby, or uninformative.

An easy way to see if the text in your application has the right personality and disposition is to see if it sounds like a friend assisting another friend with a task or objective. For example a message like “Error Code: C82FF4B696” or “Critical data not found” is likely to confuse, frustrate and even anger people. A message with the right personality and disposition might look something more like, “There is some info missing here. Please put your name in the text box to continue.”

2. Capitalization

Surface displays text in lowercase and all caps layouts in many places, but also uses title caps, where the first and last words of the phrase and all words in between are capitalized, and sentence caps where only the first word of a sentence is capitalized.

Title caps exceptions are articles (a, an, the), coordinating conjunctions (and, but, for, not, or, so, yet), and prepositions with four or fewer letters (at, for, in, into). For example, “Neon Tetras in My Fish Tank.”

Sentence caps exceptions are words that are normally capitalized in text such as proper nouns or feature names. For example, “I want to visit Mt. Rainier in the springtime.”

Maintain consistent capitalization practices to prevent disjointed or jagged reading experiences.

- a. Use lowercase for:
 - i. Content titles
 - ii. List titles
 - iii. List items
 - iv. Interface control labels
 - v. Example text that appears in search and text boxes
- b. Use sentence caps for:
 - i. Check box and radio buttons labels
 - ii. Progress indicators
 - iii. Status, notification, and explanatory text
 - iv. Toggle switches
- c. Use all caps for:
 - i. Application titles
 - ii. Dates and times
 - iii. AM or PM

3. Punctuation

Use consistent punctuation to avoid confusion, clarify ambiguous text, and provide direct emphasis as needed.

- a. Ampersand (&)
 - i. Okay to use in settings or menu lists, for example: Date & Time; Clocks & Alarms.
- b. Colon (:)
 - i. Do not use a colon at the end of labels for controls such as text boxes, drop-down menus, and progress bars.
 - ii. Do not use a colon when the text box or drop-down list is embedded in a sentence or when the drop-down list appears in the main window.

- iii. Do not use a colon at the end of group headings or column headings.
- iv. Use a colon to introduce numbers or other variables, for example:
Percent Downloaded: XX%
- c. Ellipsis (...)
 - i. Use an ellipsis in progress indicator labels to indicate a continuing action, for example, when a person downloads a file. Even if there is a visual progress indicator, you still want to use the ellipsis.
 - ii. Do not use an ellipsis in headings.
 - iii. Do not use an ellipsis in button labels.
- d. End punctuation (. ? !)
 - i. Use end punctuation only in instructional text. Do not use end punctuation if instructional text appears in a label or button.
 - ii. Do not use a period at the end of option or check box text labels, even if the label is a sentence.
 - iii. Separate sentences with one space after the ending punctuation, not with two spaces.
 - iv. End a question with a question mark. Avoid phrasing labels as questions.
- e. Parenthesis ()
 - i. Avoid using parenthesis in applications if possible, but use a parenthesis if you need to include an acronym or other short piece of information.

Section 4.0:

SURFACE HARDWARE

This section discusses design considerations and guidelines that relate specifically to the Surface hardware.

Section 4.1:

INPUT METHODS

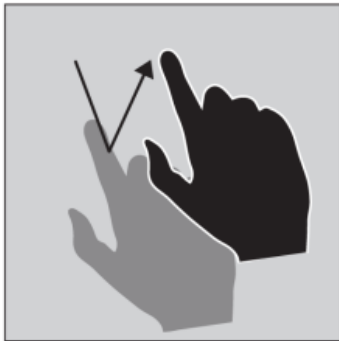
1. Vision based touch

The primary mode of interaction with Surface is touch. Surface is designed for touch from the ground up, and Surface applications should be written with direct touch interaction as the key driving force.

Fingers & blobs – Surface recognizes and differentiates between fingers and blobs. Fingers are identified when someone places their finger on the screen. The number of fingers, their position, and orientation – which direction they're pointed – are detected by the vision input system. Other items placed on the screen not identified as fingers or tags are classified as blobs. Basic size information is provided and an arbitrary orientation is assigned. Orientation values for blobs are generally not as reliable as fingers or tags.

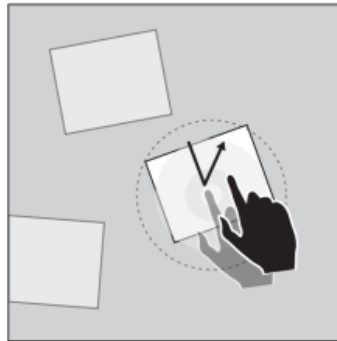
Touch interactions – The Surface SDK manipulation processor recognizes three discrete manipulations: move, rotate, & resize. The fact that there are only three manipulation gestures in the Surface SDK is a technical fact, but from an interaction perspective, there are many different direct touch interactions that a person can use with these manipulations. The following illustrations highlight the variety of touch interactions that can be performed on a virtual object with one finger or several fingers, using one or more hands.

Tap – Press and then release

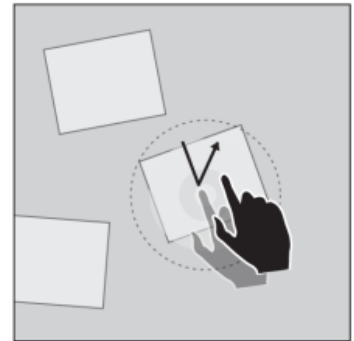


Tap

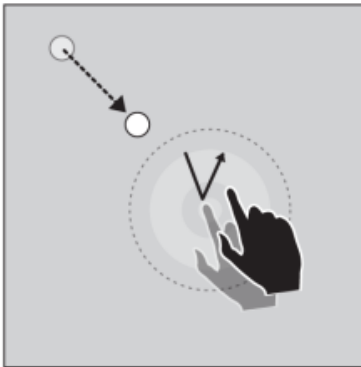
=



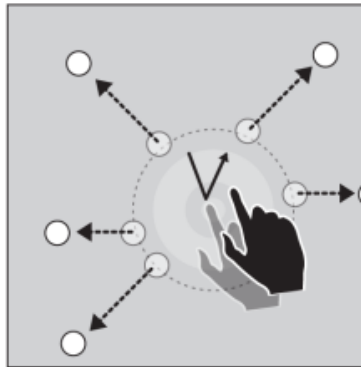
Tap to Select



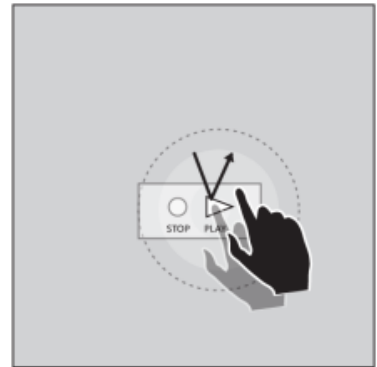
Tap to Deselect



Tap to Draw Near

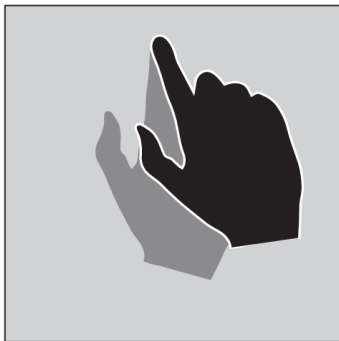


Tap to Scatter



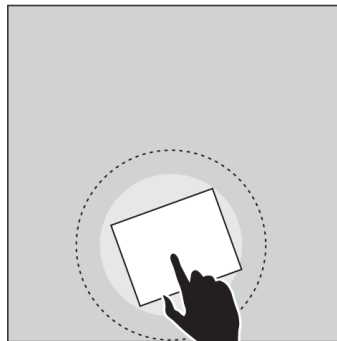
Tap to Activate Control

Hold – Press and then hold

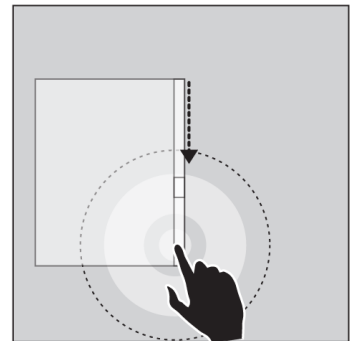


Hold

=

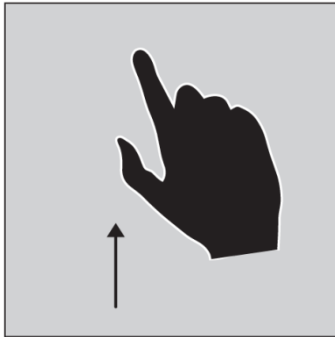


Hold to Select



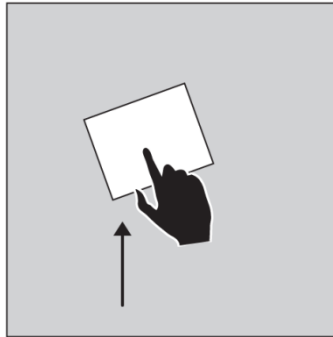
Hold to Autoscroll

Slide or push – Move the object under your finger with a sliding or pushing action

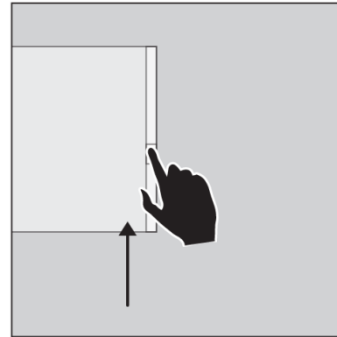


Slide or Push

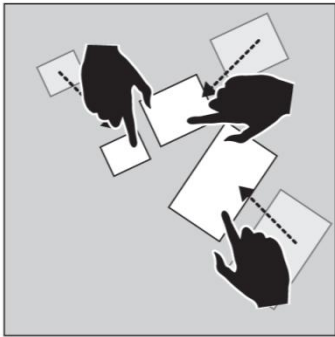
=



Slide or Push to Move

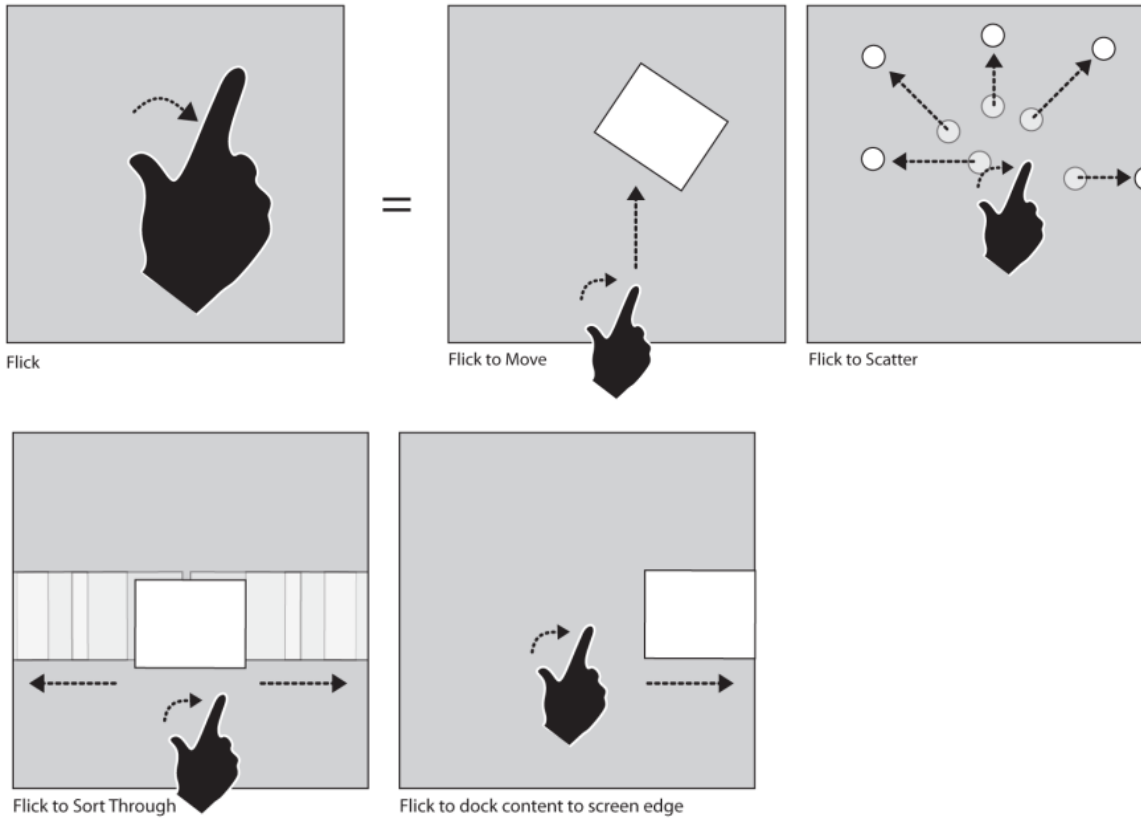


Slide or Push to Use Controls

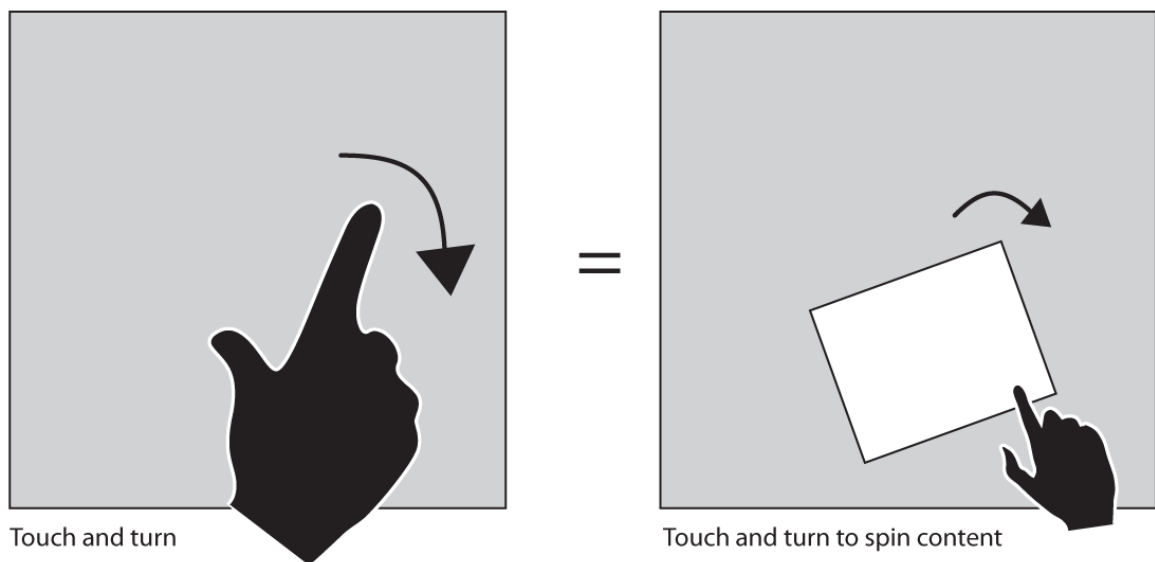


Slide or push to collect content

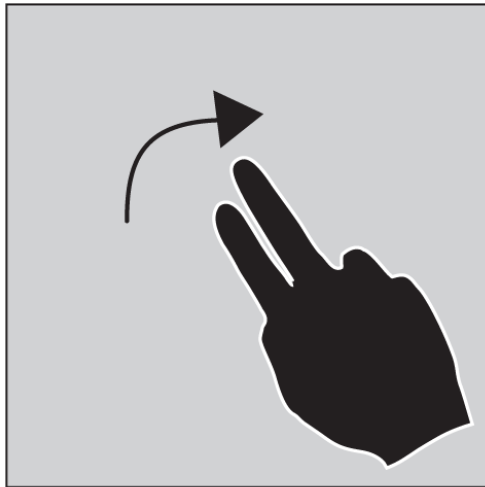
Flick – Press, slide quickly, and then release



Touch and turn – Slide your finger near the outside edge of a piece of content and rotate it around its center

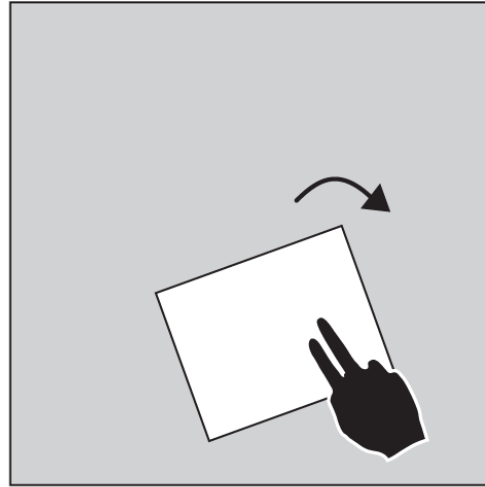


Spin – Twist quickly with two fingers to rotate the object



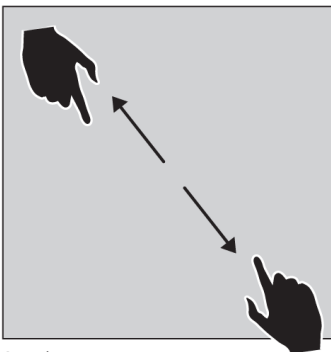
Spin

=



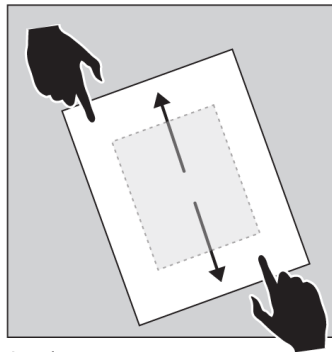
Spin to turn content

Stretch – Move fingers apart with two hands

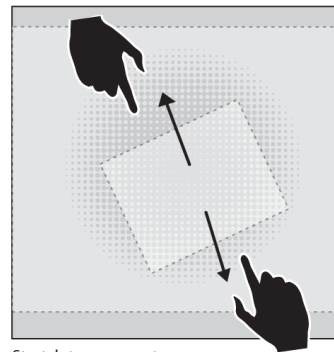


Stretch

=

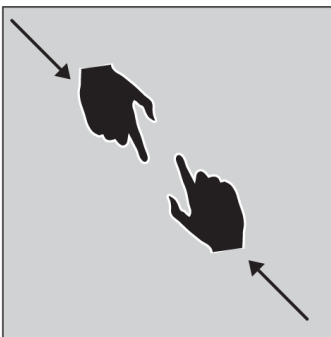


Stretch to resize



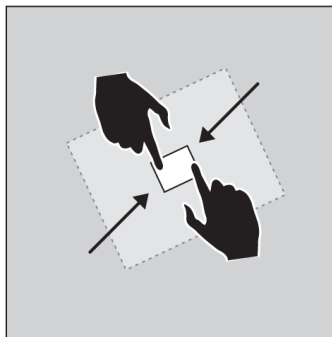
Stretch to zoom out

Shrink – Bring fingers together with two hands

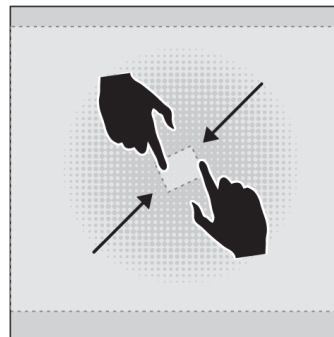


Shrink

=

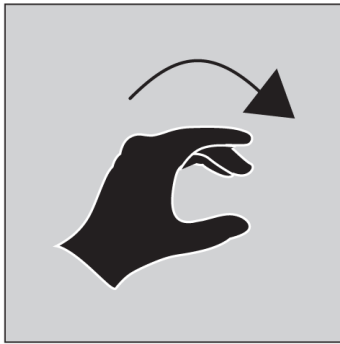


Shrink to reduce size



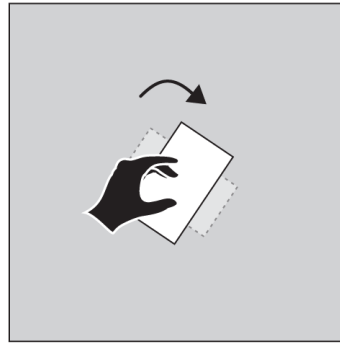
Shrink to zoom out

Twist – Twist the object with two or more fingers, like turning a knob or sheet of paper

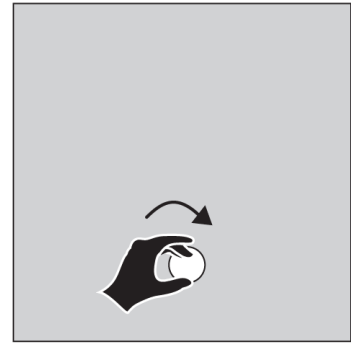


Twist

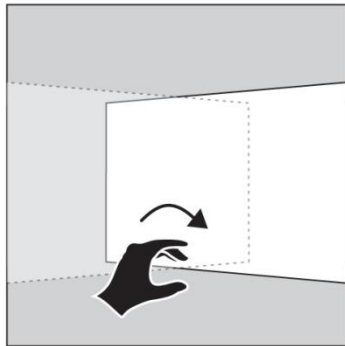
=



Twist to spin objects

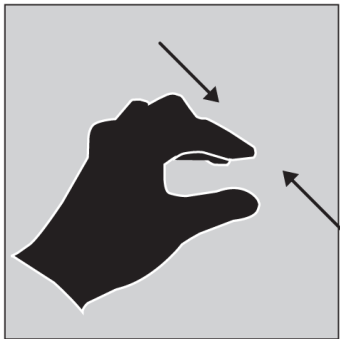


Twist to adjust control



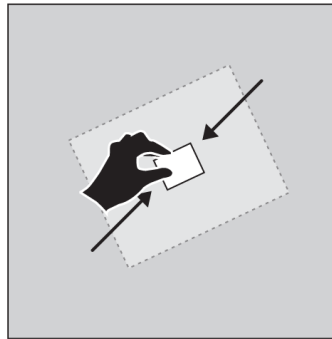
Twist to change perspective

Pinch – Move two fingers together on one hand

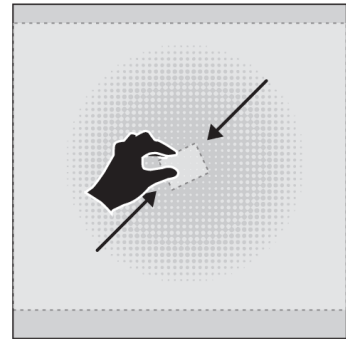


Pinch

=



Pinch to reduce size



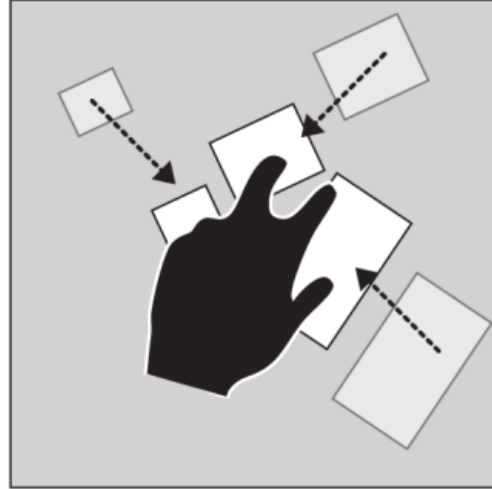
Pinch to zoom out

Squeeze – Move three or more fingers together on one hand

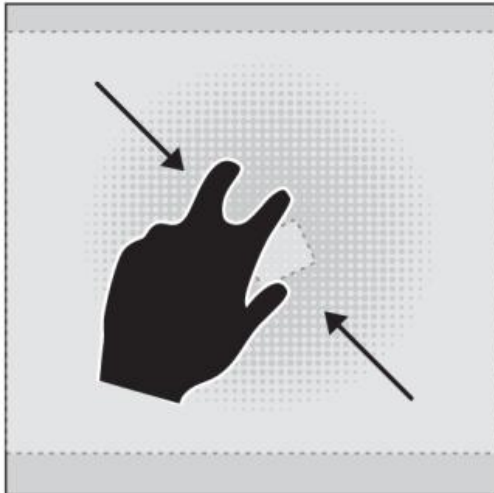


Squeeze

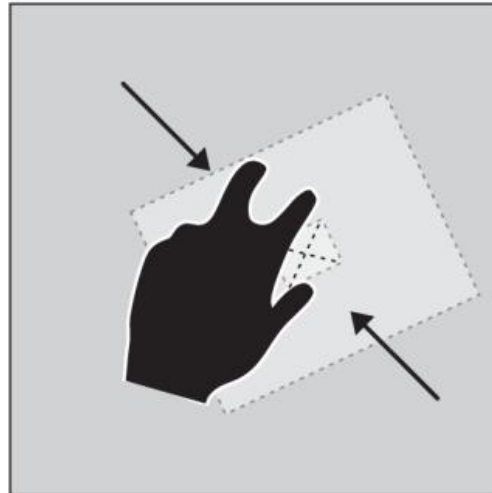
=



Squeeze to collect content

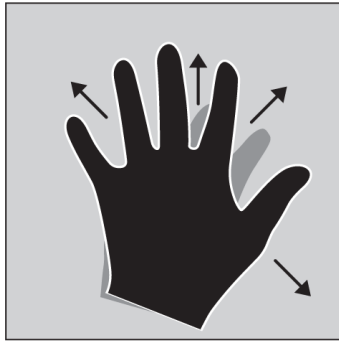


Squeeze to zoom out



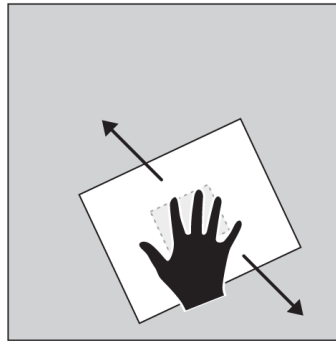
Squeeze to close content

Spread – Move fingers apart on one hand

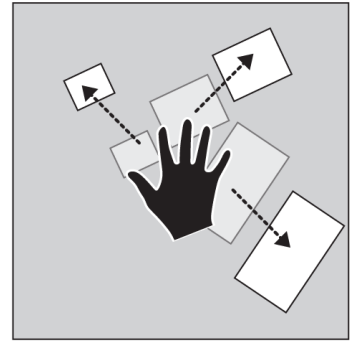


Spread

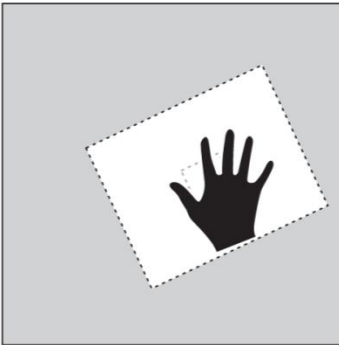
=



Spread to enlarge content

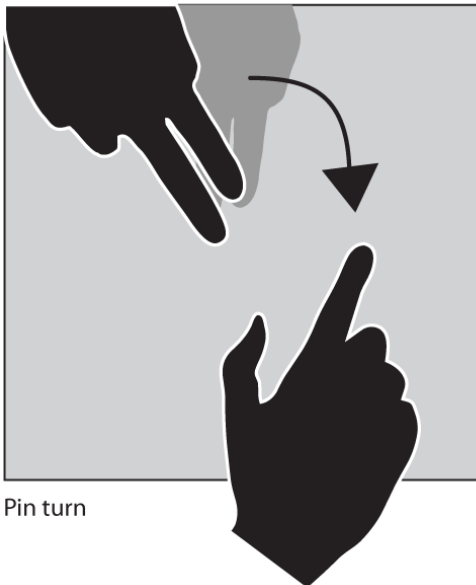


Spread to scatter content



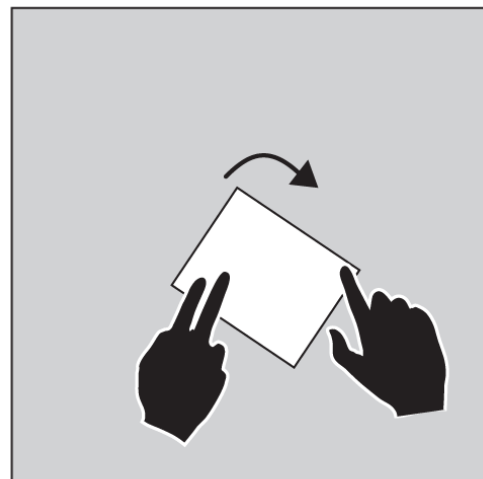
Spread to open content

Pin turn – Pin the object down with one finger while the other hand drags the object around the pinned point



Pin turn

=



Pin turn to rotate content

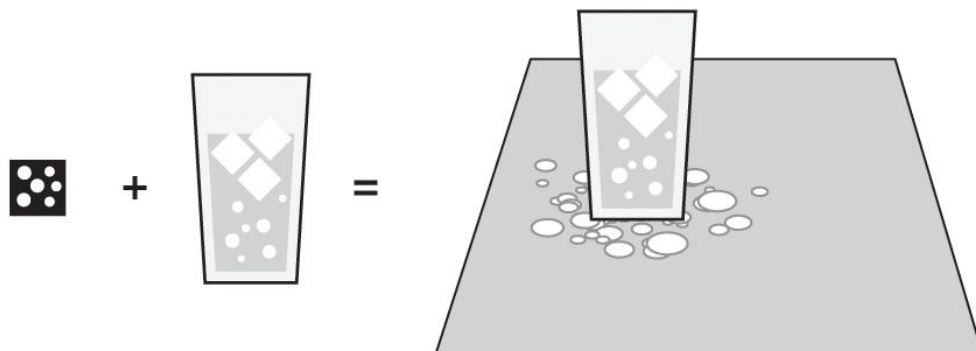
2. Objects

Surface not only supports touch interaction with fingers, it also supports a range of interactions with objects touching the screen. Using objects in Surface applications provides these unique benefits:

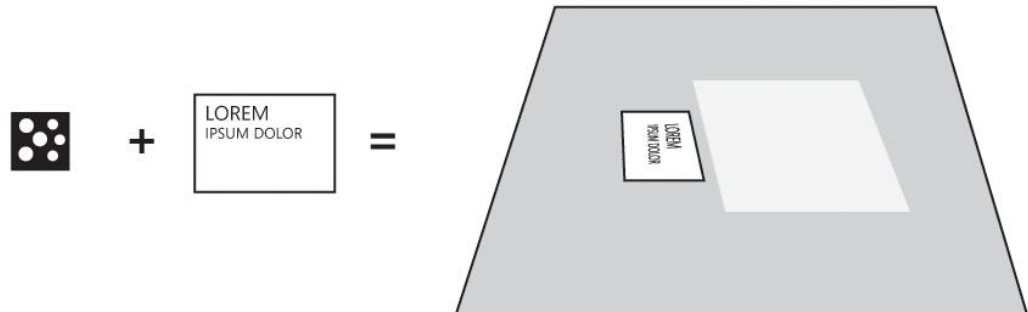
- Helps people easily share control of an application
- Helps the application identify who is using it and where they are physically located in relation to the screen
- Provides people intuitive ways to instantly understand and interact
- Provides a way to instantly link physical objects with digital content
- Provides a virtually endless number of discrete identifiers and identities
- Provides a way for people to take digital information with them
- Provides a seamless link between Surface and other mobile devices
- Creates magical, engaging experiences in ways not previously possible

The types of objects supported are explained in detail below.

Tags & orientation – A tagged object is a physical object that has a special tag with a pattern of dots affixed to it that Surface sees in infrared (IR) light wavelengths. IR light is not visible to human eyes, but Microsoft Surface hardware sees and recognizes the tag to perform the actions associated with the tag. For example, an application in a restaurant or bar could use a tag on the bottom of a glass so that when a customer puts the glass on the screen, bubbles appear around the bottom of the glass.



In another example, a tag can be used to display a menu or other information when a person places an object on the screen.



Tagged objects can also be dynamically associated with content in a particular application. This means that objects can be reused across applications without having to hardcode specific values. One application may prompt a restaurant server to place a tagged glass of orange juice down on a particular spot of the display. That application does not initially require a specific unique tag value, but the act placing that object down in a known spot lets the server identify to the application that the object placed there is a glass of orange juice. This can be repeated as needed; for example, a server might also add a tagged cup of coffee to the table in a specified spot. Once that tag is associated with the coffee, it can remain so until the customer dining experience is ended. If someone picks up the cups and places them back down on a different part of the screen, the Surface will still be able to identify them.

Object blob data, raw image & contact geometry – In addition to seeing special tags, Surface hardware can most objects placed on the screen, as long as they reflect back enough infrared light. Application developers can use the dimensions of objects which reflect basic shapes to their advantage; such as simple ellipses and rectangles. For example a coin placed on the screen with a fixed and known diameter can be easily recognized within the context of a particular application. A cube or cylinder of fixed dimensions might also be placed on the screen for a specific interaction without the need for a tag.

More advanced applications can use the raw image data provided by the vision input system to analyze contact geometry, the outline or silhouette of an object, to provide meaningful interaction with complex shapes. For example a developer can

use the outline of a human hand to create new types of direct touch interactions. Objects with unique shapes like stars, triangles, octagons and so forth could also provide special interactions without the need for tags. This type of object recognition does come with potential application development and performance costs, but is capable of providing experiences well beyond those of typical Surface applications.

Some Surface applications use the raw bitmap image from the sensors to identify objects and their purpose within the context of a particular application. This method generally differs because it goes beyond looking at basic shapes to identify an object. Using raw image may be a good method to capture a snapshot of an object and compare it against a database of previously captured images. This method can provide positive identification for much more complex objects.

Other applications may not even be concerned with identifying a complex object at all, but may instead use the vision input processors to capture an image, much like a photocopier or scanner, so that people using the Surface may view, edit, and interact with the image. This type of image capture can be limited if objects don't reflect a sufficient amount of infrared light. Application developers and designers should experiment with the properties of their objects and test them in real deployment locations early in the planning phase of development.

Transparent objects – Many objects on Surface literally light up when placed on the screen. Some of these objects contain clear tags on them which contain the same byte patterns as standard tags, but can't be seen by the human eye. This enables clear glass and plastic objects to be placed on the screen and have software driven images projected right through them.

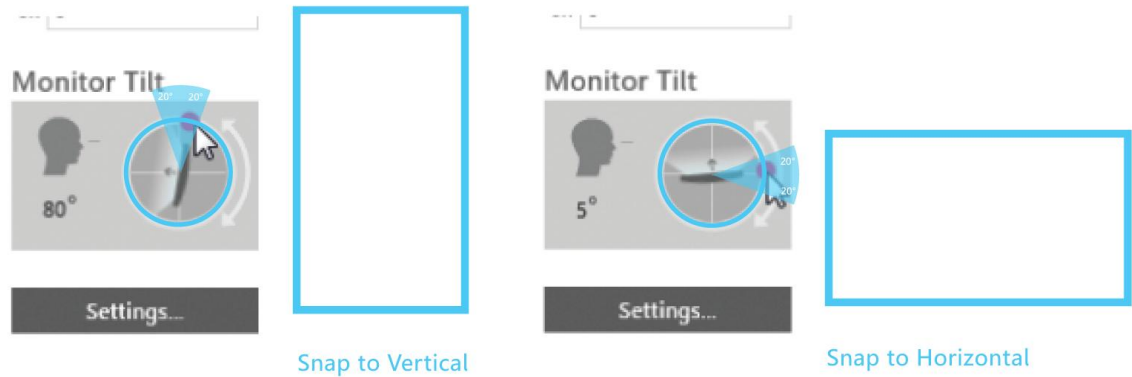
This can transform an inert plastic cube into a lively token, or a dinner glass into an interactive control object. Specialized objects can even use unique properties to alert the Surface when the liquid level in a glass has reached a certain level, or project animated visuals through objects using fiber optic elements.

These types of objects are often inexpensive ways to create unique experiences that other devices and platforms cannot deliver. The possibilities for using objects to interact with Surface applications are virtually limitless.

Section 4.2:

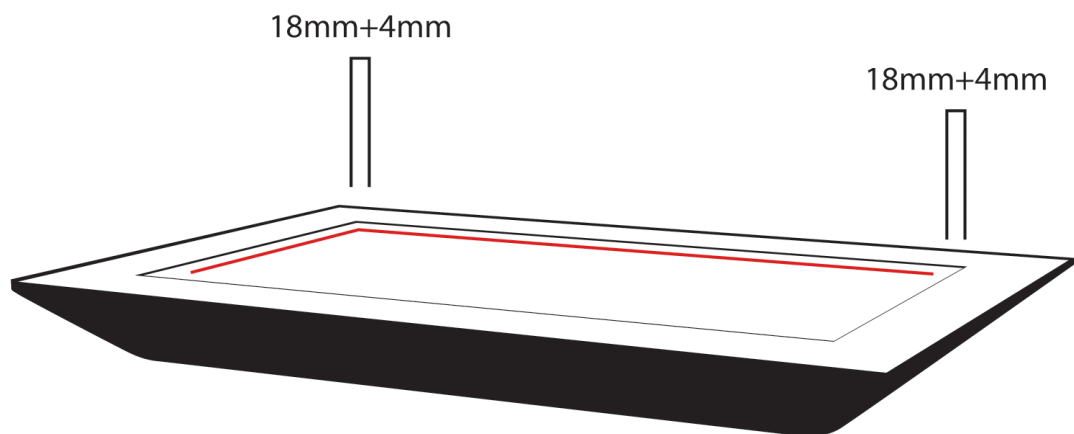
HARDWARE SPECIFIC CONSIDERATIONS

1. Tilt



Microsoft Surface hardware is designed for deployment in horizontal and vertical deployments. Sensors in the device will indicate the angle of tilt that the device is currently at. A horizontal device will report back a value close to 0° degrees and a vertical device will report back a value close to 90° degrees. Some devices may be mounted at an incline between 0° degrees and 90° degrees so it's important for application developers to keep this in mind when designing and testing their applications.

2. Bezel buffer



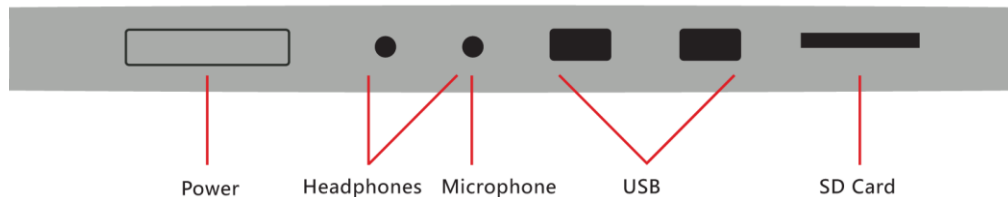
Application content should be designed with Surface hardware parameters in mind. The minimum size for touchable content should be at least 18mm x 18mm. Touchable content displayed at the screen edge should be inset at least 4mm to

ensure that finger and other contacts will be recognized by the sensors. The Surface Access Points in their rest state provide a good visual indicator of the minimum touch target size and distance to display interactive content away from the screen edge. Content docked at the screen edge should stick out at least 22mm from the screen edge in order to ensure that it is usable.

3. Connectivity

Surface hardware is equipped with Bluetooth and WiFi capabilities. This enables Surface to easily connect to a wireless network and interact with other devices. Surface may also be wired directly through its Ethernet port, which provides advance remote administration capabilities.

Many Surface applications also provide connectivity with mobile devices by using cloud based services and mobile applications. When combined with the vision input and tracking capabilities of the Surface hardware the interaction with mobile devices can be quite seamless. Surface is also equipped with HDMI in and out capabilities, as well as headphone, microphone, SD card and USB ports.





Code Metrics Values

Visual Studio 2010

Code metrics is a set of software measures that provide developers better insight into the code they are developing. By taking advantage of code metrics, developers can understand which types and/or methods should be reworked or more thoroughly tested. Development teams can identify potential risks, understand the current state of a project, and track progress during software development.

Software Measurements

The following list shows the code metrics results that Visual Studio calculates:

- **Maintainability Index** – Calculates an index value between 0 and 100 that represents the relative ease of maintaining the code. A high value means better maintainability. Color coded ratings can be used to quickly identify trouble spots in your code. A green rating is between 20 and 100 and indicates that the code has good maintainability. A yellow rating is between 10 and 19 and indicates that the code is moderately maintainable. A red rating is a rating between 0 and 9 and indicates low maintainability.
- **Cyclomatic Complexity** – Measures the structural complexity of the code. It is created by calculating the number of different code paths in the flow of the program. A program that has complex control flow will require more tests to achieve good code coverage and will be less maintainable.

Note

In some cases, the calculation of the cyclomatic complexity for a method in Visual Studio 2010 differs from earlier versions. For more information, see the "Changes in Visual Studio 2010 code complexity calculations section" of [Troubleshooting Code Metrics Issues](#).

- **Depth of Inheritance** – Indicates the number of class definitions that extend to the root of the class hierarchy. The deeper the hierarchy the more difficult it might be to understand where particular methods and fields are defined or/and redefined.
- **Class Coupling** – Measures the coupling to unique classes through parameters, local variables, return types, method calls, generic or template instantiations, base classes, interface implementations, fields defined on external types, and attribute decoration. Good software design dictates that types and methods should have high cohesion and low coupling. High coupling indicates a design that is difficult to reuse and maintain because of its many interdependencies on other types.
- **Lines of Code** – Indicates the approximate number of lines in the code. The count is based on the IL code and is therefore not the exact number of lines in the source code file. A very high count might indicate that a type or method is trying to do too much work and should be split up. It might also indicate that the type or method might be hard to maintain.

Anonymous Methods

An *anonymous method* is just a method that has no name. Anonymous methods are most frequently used to pass a code block as a delegate parameter. Metrics results for an anonymous method that is declared in a member, such as a method or accessor, are associated with the member that declares the method. They are not associated with the member that calls the method.

For more information about how Code Metrics treats anonymous methods, see [Anonymous](#)

[Methods and Code Analysis.](#)

Generated Code

Some software tools and compilers generate code that is added to a project and that the project developer either does not see or should not change. Mostly, Code Metrics ignores generated code when it calculates the metrics values. This enables the metrics values to reflect what the developer can see and change.

Code generated for Windows forms is not ignored, because it is code that the developer can see and change.


See Also

Other Resources

[Measuring Complexity and Maintainability of Managed Code](#)

Community Content

© 2011 Microsoft. All rights reserved.

The perspective wall: detail and context smoothly integratedFull Text:  [pdf](#)

Authors: [Jock D. Mackinlay](#) [Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA](#)
[George G. Robertson](#) [Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA](#)
[Stuart K. Card](#) [Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA](#)

Published in:

- Proceeding
CHI '91 Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology

[ACM](#) New York, NY, USA ©1991[table of contents](#) ISBN:0-89791-383-3 doi>[10.1145/108844.108870](#)

1991 Article

**Bibliometrics**

- Downloads (6 Weeks): 16
- Downloads (12 Months): 175
- Citation Count: 196

Tools and Resources[Request Permissions](#)

TOC Service:

[Email](#) [RSS](#)[Save to Binder](#)

Export Formats:

[BibTeX](#) [EndNote](#) [ACM Ref](#)

Upcoming Conference:

[CHI '12](#)

Share:



Tags: [animation](#) [design](#) [human factors](#) [human factors](#) [interaction](#) [styles](#) [theory](#)

Powered by **THE ACM GUIDE TO COMPUTING LITERATURE**

The ACM Digital Library is published by the Association for Computing Machinery. Copyright © 2011 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)