

Exploring the roles of Scrum Master & Product Owner

Auction.com



David Luzquinos | dluzquinos@luzquinos.com | @_davidl_



- Agile & Lean Consultant and Coach
- Over 15 years experience across spectrum of industries including IT, financial, travel, data marketing, gaming, publishing and manufacturing
- Participated in a variety of roles from developer to Project/Program/Product to Director
- Expertise in applying Agile & Lean approaches since 2004
- Focus in helping companies achieve organizational agility beyond development teams



Objective

- To learn about Agile
- To provide you with the context to start or continue your journey
- **ONLY EXPERIENCE, CONTINUOUS LEARNING, COLLABORATION, SHARING AND PERSISTENCE WILL PROVIDE YOU WHAT YOU NEED**



Not just how, but why



Organizing Tools

- Parking Lot
- How about some Ground Rules:
 - Be on time; we will start and end on time
 - One conversation at a time
 - Electronics used only by exception
 - Regular breaks
 - Others? What will improve your workshop experience?



Learning Objective

- What's one learning objective YOU have for the workshop today?



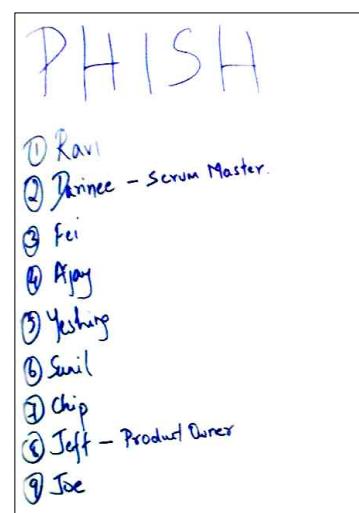
Focus Areas

- Review Roles:
 - Scrum Master
 - Team
 - Product Owner
- Envisioning
- User Roles
- User Experience Mapping
- Release Planning
- User Stories
- Splitting Stories
- Prioritization



Scrum Simulation: Form Your Teams

1. Form Teams
 - Try to establish a team with cross-functional roles
2. Find a Product Owner
3. Create Team Name
4. List Member Names
5. Choose Scrum Master



Scrum Simulation: Overview

Sprint Goal: Develop a mailing brochure in one sprint

- Complete Sprint Planning Meeting (15 min.)
- Conduct Sprint (30 min.)
- Conduct a Sprint Review and Retrospective (14 min.)
- *Make it something creative and fun!*

Adapted from work by Jean Tabaka



Scrum Simulation: Sprint Planning Meeting

1. Product Owner selects 3 – 5 product features
2. Product Owner puts features in priority order
3. Team decides on the goal of the sprint
4. Team decomposes features into 2 to 3 tasks per feature
5. Team has 24 minutes per team member of capacity; multiply # of team members times 24 to get total team capacity
6. The team members estimate tasks
7. Add up all task estimates and commit to delivery based on capacity versus total of task estimates

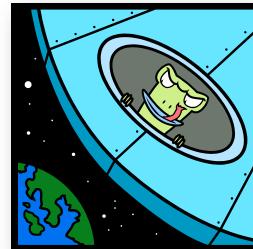
Adapted from work by Jean Tabaka



Scrum Simulation: Product Backlog

- Create cover art, brand, and/or logo
- Define major topics for Martian tourism
- Describe “Art Interests in Europe” tour
- Describe a tour based on photosynthesis
- Outline a “7 wonders of the world” expedition
- Set prices for the tours
- Outline warning messages (gravity, oxygen, fungi, etc.)
- Suggest clothing options
- Explain travel options to/from Mars
- Describe a “Human Sports” tour
- Outline refund policy
- Suggest related services
- Define advertisers
- Define a 12-month campaign
- Set-up how to get more information

Adapted from work by Jean Tabaka



Scrum Simulation: Sprint Plan



Scrum Simulation: The Sprint

- Conduct sprint (30 min.)
- Three 8-minute days in sprint
 - Day 1 (8 min.)
 - Daily Scrum (3 min.)
 - Day 2 (8 min.)
 - Daily Scrum (3 min.)
 - Day 3 (8 min.)

Daily Scrum – 3 min.
• Done since last meeting
• Plan for today
• Impediments?



Imagery: Copyright © 2012, Kenneth S. Rubin and Innolution LLC



Scrum Simulation: Sprint Review

- Conduct Sprint Review and Retrospective - 14 minutes
 - Sprint Review (7 min.)
 - Team demonstrates brochure to Product Owner
 - Collect feedback from Demo
 - Sprint Retrospective (7 min.)
 - Scrum Master facilitates team in Retrospective
 - Use “Pluses and Deltas” technique – Capture “What went well” on Pluses first and then capture “What could be improved” on Deltas

+	▲
* What went well	* What could be improved



The Team Review



Scrum Teams Are Compact

- Seven plus or minus two
- Preferably co-located
- Go fast through face-to-face communication
- Cross-functional with flexible roles
- Scrum scales by adding teams, not increasing team size

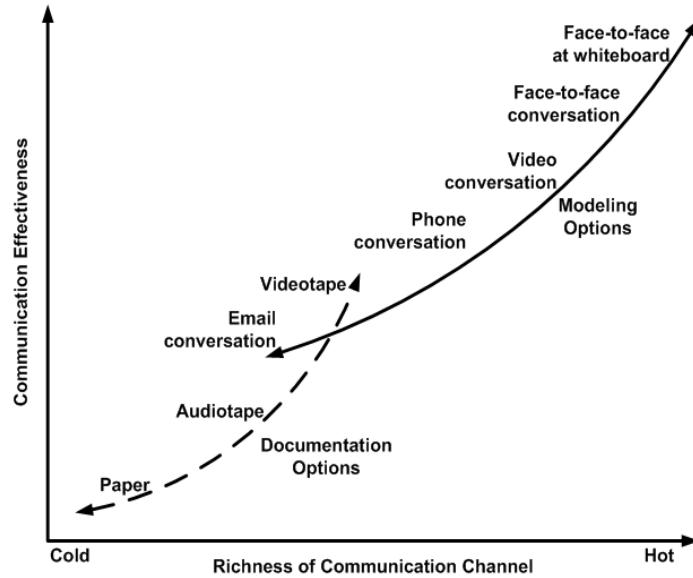


Scrum Teams Self-Organize

- Self-disciplined
- Team makes decisions collaboratively
- Cooperative development, role-sharing
- Each member is accountable for the team meeting its sprint goal
- Share leadership roles/responsibilities
- Create a quality product
- Communicate progress and impediments



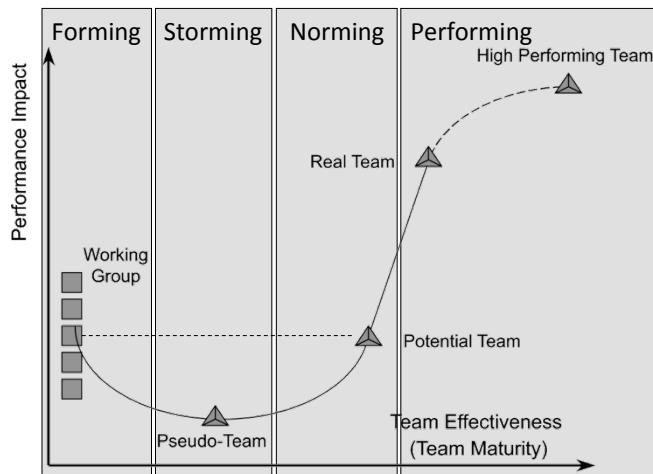
Scrum Teams Use High-Bandwidth Communication



Copyright 2002-2005 Scott W. Ambler
Original Diagram Copyright 2002 Alistair Cockburn



The "Tuckman" Model



Source: *The Wisdom of Teams: Creating the High-Performance Organization* by Jon R. Katzenbach and Douglas K. Smith



High Performing Teams

- Built-in Instability
- Self-Organizing Project Teams
- Overlapping Development Phases
- Multi-Learning
- Subtle Control
- Organizational Transfer of Learning



Takeuchi, Nonaka, "The New New Product Development Game", HBR, Jan-Feb 1986
<http://www.kingscourtgalleries.co.uk/artwork/Rugby.jpg>



Team Development Discussion

- In your teams, discuss each of the stages through performing
- How might the Scrum Master help the Team through each of these stages?

15 minutes



Scrum Master Review



Scrum Master

- Is a servant leader, bulldozer, coach, change agent
- Champions, facilitates, and protects the Scrum process
- Removes impediments and prevents interference for the team
- Insures information created by team and their efforts are easily viewed by all with an interest
- Helps team identify area for potential improvements and implement solutions
- Focus is on building effective “Team”



Why not Agile Project Manager?

“I wanted to highlight the extent to which the responsibilities of the Scrum Master are different from those of a traditional project Manager”

Ken Schwaber
Agile Project Management with Scrum



Scrum Master and the Team

- Servant Leader to Product Owner and Team; nobody reports to the Scrum Master
- Encouraging disciplined engineering practices
- Removes impediments or barriers
- Enables cooperation
- Facilitator
- Focus is successful Team



Impediments

- Impediment: anything that slows down, distracts, interrupts the team or a team member while performing activities aiming to reach the Sprint goal
 - Personal Impediment
 - Team Impediment
 - Organizational Impediment
- A good practice for a Scrum Master is to keep an impediment backlog and report on the status of the impediments at the Daily Scrum



Scrum Master and the Organization

- Shields the Team from distractions or interruptions
- Champion for the Scrum process teaching as a simple process with few rules
- Change agent
- Radiates information
- Advocate best practices

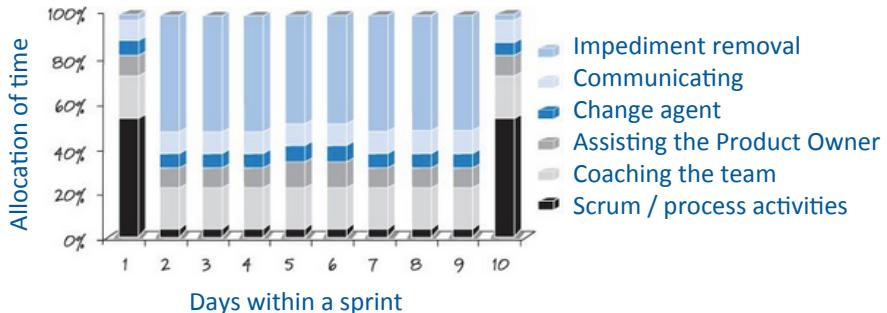


Scrum Master and the Organization

- Optimal for the Scrum Master to reside within an organization
- Optimal for the Scrum Master to not also be a member of the team (developer, tester, etc.)
- One of the common tasks that are easy for new Scrum Masters to grasp is the removal of impediments
- However, what is often overlooked, is the people aspect



A Day in the life of a Scrum Master



Source: Essential Scrum, Kenneth S Rubin



Tools for the Scrum Master

Active listening
Explaining
Questioning?
Safe to fail
Exposing Consequences
Participating
Team Agreements
Challenging!
Observing
Experiments
Observing



Example Mapping of Agile Roles

Area	PO	SM	Team	Other Mgr
Integration	✓			?
Scope	Macro-level		Sprint-level	
Time	Macro-level		Sprint-level	
Cost	✓		Story/task estimating	
Quality	✓	✓	✓	✓
Team			✓	Formation
Communication	✓	✓	✓	✓
Risk	✓	✓	✓	✓
Procurement	✓			✓

Source: Kenneth S Rubin



Group Discussion – Daily Scrum

You're the Scrum Master:

You notice the scrum is taking increasingly more time. It is now more than 30 minutes long.

You notice people showing up late or not attending every Scrum

Team members are coming un-prepared to the Scrum

The team wants to email each other their Scrum updates instead of having the Scrum

A developer gives an update at a scrum that she is helping out a manager outside the team on a small assignment.

You notice that on days that you are out of the office, the Scrum does not take place

Why is this a problem?

What do you do?



Feedback Loops

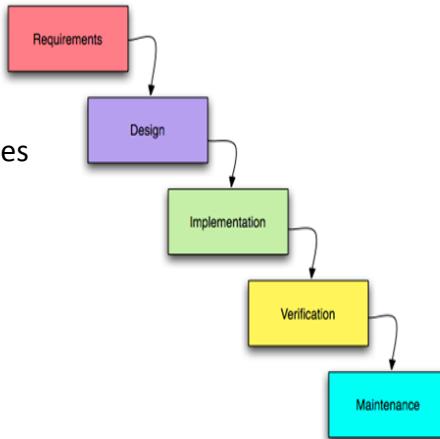


Traditional Project Management

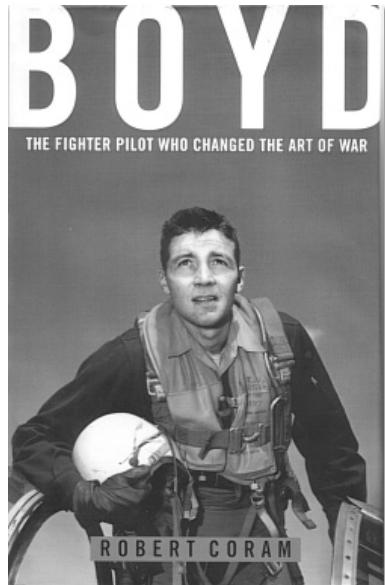
"I believe in this concept but the implementation is risky and invites failure..."

Dr. Winston Royce

Managing the Development of Large Software Systems (1970)



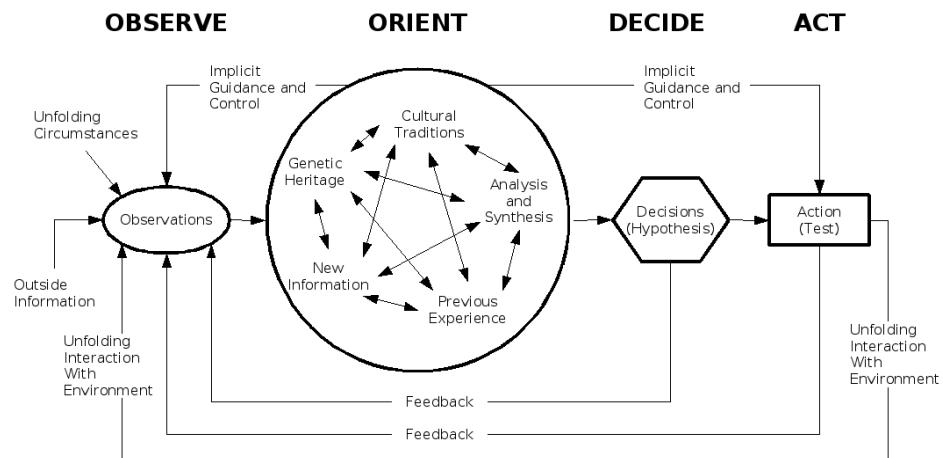
Colonel John Boyd, US Air Force



- Architect of the Desert Storm Campaign
- '40 Second' Boyd
- Creator of the OODA Loop



Colonel Boyd's OODA Loop



What do OODA Loops have to do with product development?

- One of our tasks as Scrum Masters is to manage risk
- Feedback loops provide us with a powerful tool to manage risk
- What do we want feedback on?
 - What is the quality of our product
 - Does our product meet the customer problem or needs
 - Are we on track with building the product within a reasonable timeframe and within our budget
 - Is it a viable product (*does it work?*)



Our Challenge

Increase the efficiency of our processes
and
Increase the quality of our deliverables

What do OODA loops
have to do with Quality?



Assuring Quality

How do we effectively assure Quality?

We must manage building Quality into
the Product.



Rapid Feedback Loops Assures Quality

If you want to You have to

go fast...have quality



Discussion

In a Scrum team, who owns quality?

How does the team enable quality?



Our Travel Guide Site User Stories

As a Martian I want vacation packages so that I can easily plan my trip

As a Martian I want tips on local culture so that I don't get misunderstood

As a Martian I want estimates of how much time to expect at events so that I can plan effectively

As a Martian I want a price list so that I can know how much activities cost

As a Martian I want detailed pictures of locations so that I can choose which places to visit

As a Martian I want recommended restaurants so that I can eat

As a Martian I want to look at maps so that I can find my way

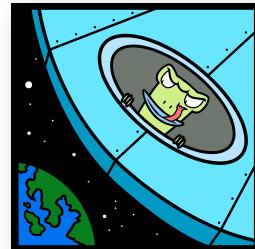
As a Martian I want a list of potential travel risks so that I survive my travel adventures

As a Martian I want to call someone when I get lost so that I get back on track quickly



Exercise: Estimate Stories

- Estimate the highest priority stories on Product Backlog
- Use Planning Poker and 1,2,3,5 point scale
- Product Owner copies stories from Backlog and presents them to Team
- Team estimates each story as it is delivered with Product Owner answering questions
- Scrum Master tabulates estimates
- Instructors work with Product Owners to coordinate activities
- Stories may be estimated by more than one team



Exercise: Complete Part 1 of Planning

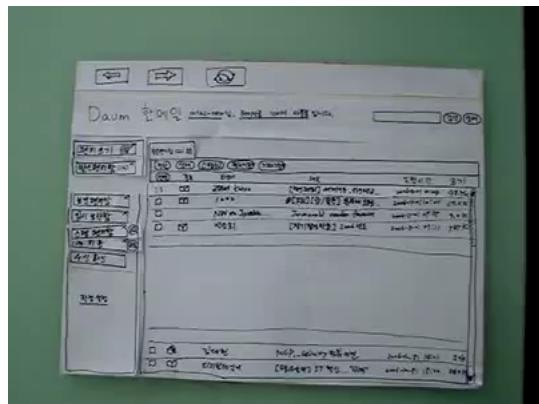
- Identify an Integration strategy
- Break out into teams
- Determine stories to be completed by each team

Definition of Done

- Paper prototype of Site
- Sections Identified and Included
- Advertising (Allocated or Placeholders)
- Fully interactive



Paper Prototyping



Exercise: Complete Part 2 of Planning

- Breakout and complete Sprint Planning as teams
- Work amongst teams
 - Clarify work
 - Identify and Understand Dependencies
 - Consider integration strategies
 - Trade Stories
 - Resolve shared member issues
 - Commit initially as single team
- Regroup
 - Teams share backlogs
 - Validate plans
 - Determine integration strategy
 - Multi-team commitment



Build the Travel Guide Web Site!

- Conduct Day 1 - 8 min
- Conduct a Daily Meeting - 3 min
- Conduct Day 2 - 8 min
- Conduct a Daily Meeting - 3 min
- Conduct Day 3 - 8 min

Remember: We want a fully integrated potentially shippable product increment at end of Sprint that meets our Definition of Done



Agile Planning



“Plans are of little importance,
but planning is essential”

- Winston Churchill



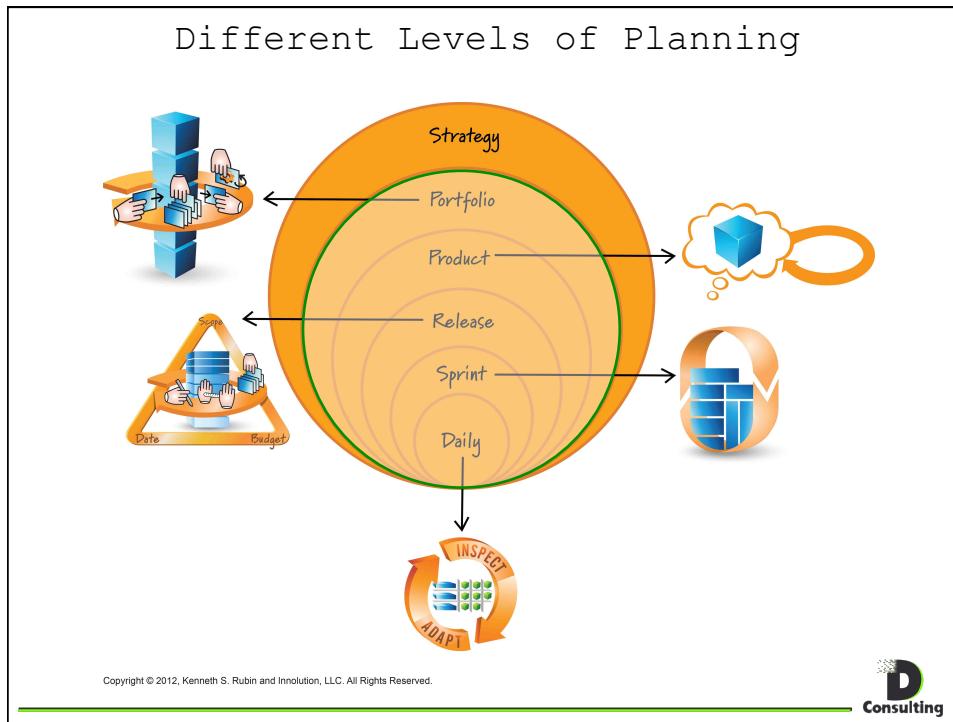
Agile Planning Principles

Scrum Planning Principles

- Can't get the plans right upfront
- Upfront planning should be helpful without being excessive
- Keep planning options open until last responsible moment
- Favor frequently replanning over conformance to plan
- Correctly manage planning inventory
- Smaller and more frequent releases
- Plan to learn fast and pivot when necessary

Source: Essential Scrum, Kenneth Rubin



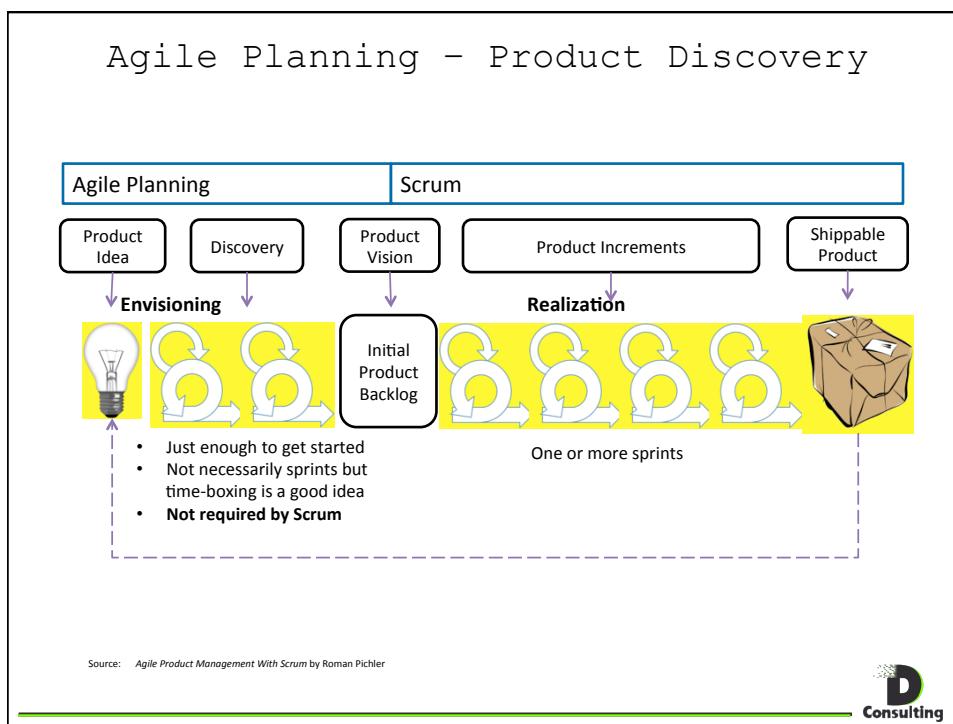
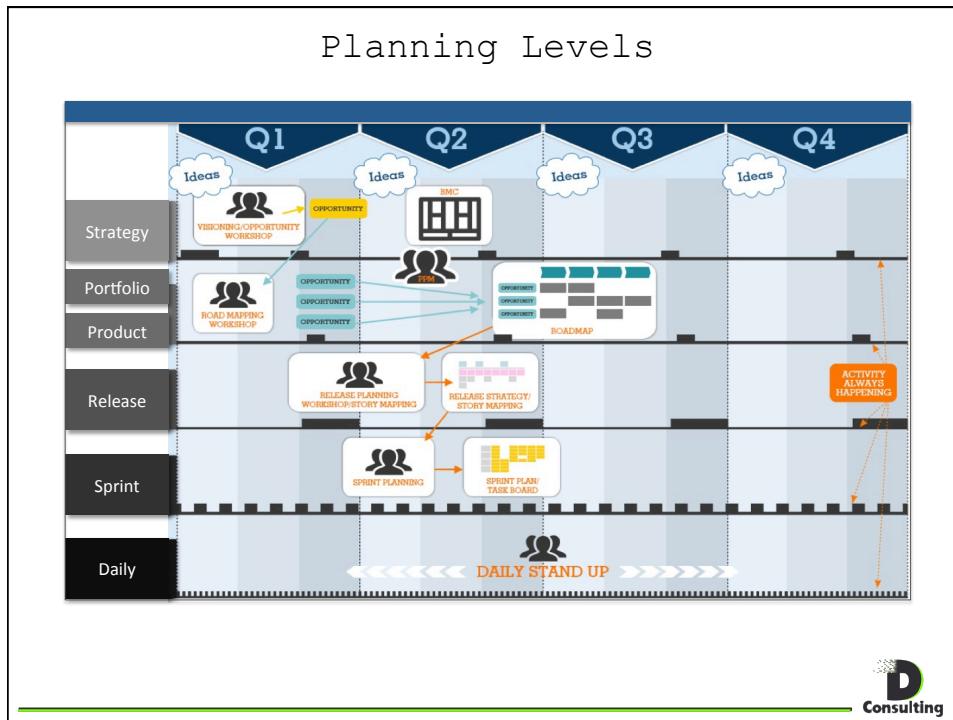


Planning Level Details

Level	Horizon	Who	Focus	Deliverables
Portfolio	Possibly a year or more	Stakeholders and Product Managers	Managing a portfolio of products	Portfolio backlog and collection of in-process products, Business Model Canvas
Product (envisioning)	Up to many months or longer	Product Managers; Stakeholders	Vision and product evolution over time	Product vision, roadmap, and high-level features
Release	Three (or fewer) to nine months	Entire Agile team stakeholders	Continuously balance customer value and overall quality against the constraints of scope, schedule and budget	User experience / Story map, Release plan
Sprint	Every iteration	Entire Agile team	What features to deliver in the next sprint	Sprint goal and sprint backlog
Daily	Every day	Scrum Master; development team	How to complete committed features	Inspection of progress and adaptation of how best to organize the upcoming days work

Source: Adapted from Essential Scrum, Kenneth Rubin

D Consulting



A week in the life of Product Discovery

Mon	Tue	Wed	Thu	Fri
morning standup	morning standup	morning standup	morning standup	morning standup
Kick-off + stakeholders identify product goals	Story Map - high level whole product	user interviews and observation Architect and developer breakouts	Design Studio Solution Workshop	Sizing and release planning
Build customer profiles and pragmatic personas	Story Map deep dive - most critical product areas	Story Map deep-dive - problems and questions	UI storyboard refinement Engineering prototypes	Tighten up plan for presentation Final plan review & green light or weekly outcome review
share-out	share-out	share-out	share-out	

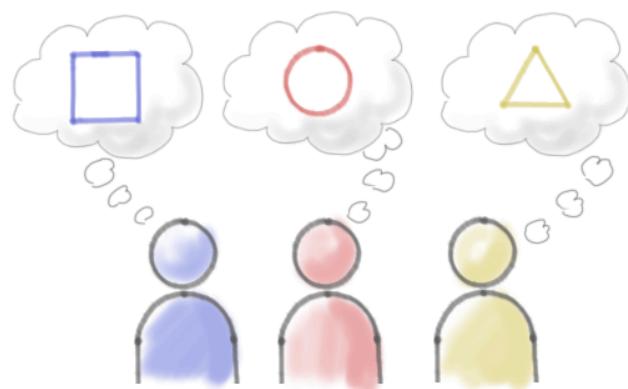
- Keep visibility high inside the product team with daily huddles
- Stay transparent with visible output and weekly outcome reviews



Envisioning



Building Shared Understanding



"I'm glad we all agree then"

www.AgileProductDesign.com, Jeff Patton



Building Shared Understanding



www.AgileProductDesign.com, Jeff Patton



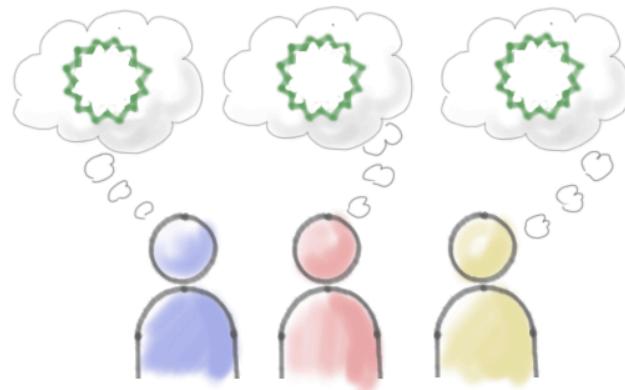
Building Shared Understanding



www.AgileProductDesign.com, Jeff Patton



Building Shared Understanding



www.AgileProductDesign.com, Jeff Patton



Product Vision

- Acts as the overall goal
- A sketch of the product at a coarse-grained level
- Communicates the essence of the product
- A shared goal that provides direction but broad enough to facilitate creativity



Source: *Agile Product Management With Scrum* by Roman Pichler



Questions to consider in the vision

- ✓ Who is going to buy the product?
- ✓ Who are the users?
- ✓ Which customer needs will the product address?
- ✓ Which product attributes are critical to address the customer needs?
- ✓ How does the product compare against existing products?
- ✓ What are the product's unique selling points?
- ✓ What is its target price?
- ✓ What is the target timeframe and budget to develop and launch the product?

Source: *Agile Product Management With Scrum* by Roman Pichler



Some tools to help create a unifying vision

- Elevator statements
- Business Model Canvas
 - Assists organizations in lining up their activities by illustrating potential trade-offs
- Trade journal, press release
 - What would you like to read about the product
- Product Box
- Prototypes and mock-ups
 - Helps you understand the product
- Personas and scenarios
 - Name some users and describe how the product will influence their life
- Product roadmap and release plan



iPod Vision – Elevator Pitch

“In your pocket”

The iPod will be a portable digital music player that will hold 5000 songs. It will have a battery life measured in days, not hours. You will navigate the thousands of songs with a single finger. You will sync all your music from your computer to the iPod in minutes automatically, so you can have all your music in your pocket.



One Type of Elevator Pitch template

FOR <target customer>
WHO <statement of the need>
THE <product name>
IS A <product category>
THAT <key benefit>
UNLIKE <primary competitor>
OUR PRODUCT <further
differentiation>

(From Geoffrey Moore, Crossing the Chasm)



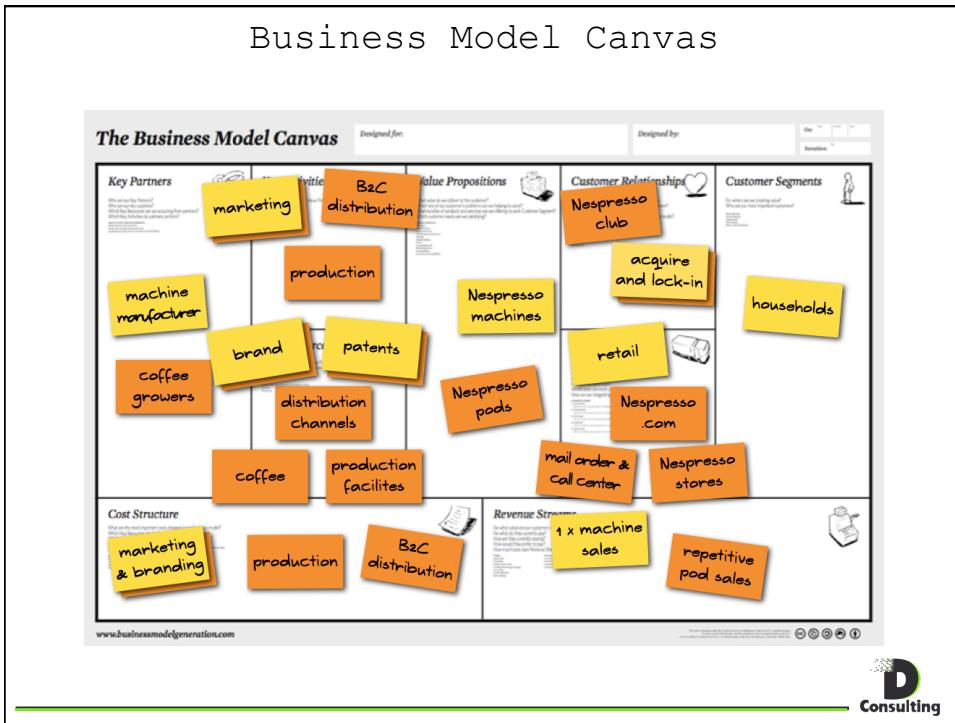
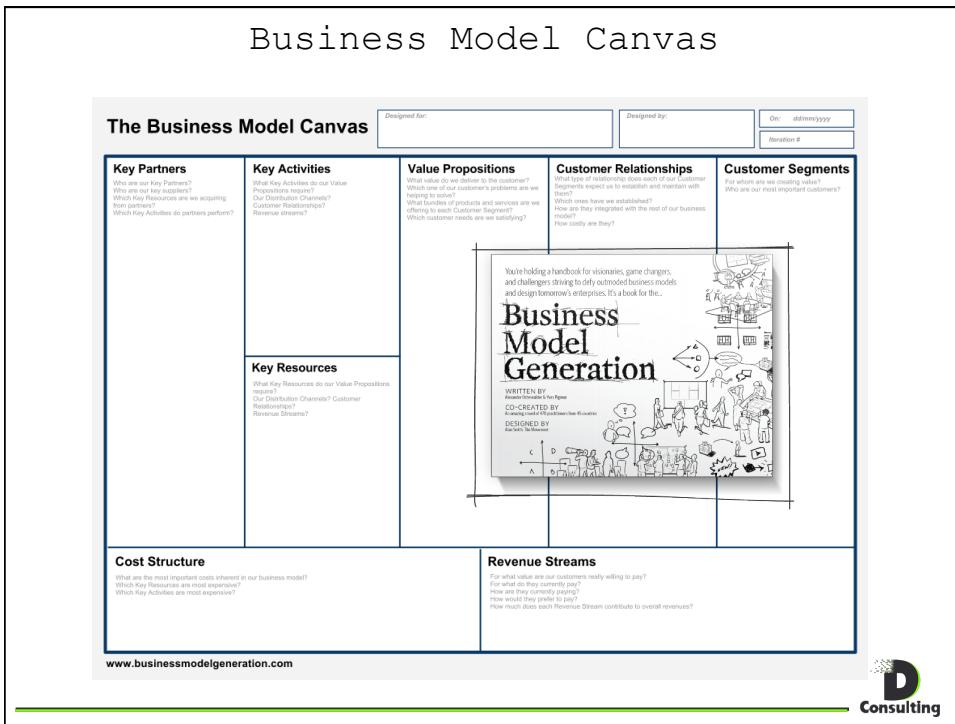
Example elevator statement

Elevator pitch example from Geoffrey Moore's
"Crossing the Chasm Book":

Silicon graphics in Hollywood:

For post-production film engineers
Who are dissatisfied with the limitations of traditional film editors
Our workstation is a digital film editor
That lets you modify film images any way you choose.
Unlike workstations from Sun, HP, or IBM,
We have assembled all the interfaces needed for post-production film
editing.





Press Release

FIGHT THE HEAT, not the habit!

Zotos Professional introduces new Flat Iron Addict, for maximum protection from habitual flat ironing

The addiction began with Blow Dry Addict, a damage-diminishing crème for the blowout-obsessed. Now, Zotos Professional introduces **Flatiron Addict**, the next fix in its Addict line for styling junkies who are hooked on hot tools.

The perfect prescription for anyone who incessantly irons their hair, this fine-mist, fast-drying spray lets you indulge your addiction by **shielding hair against damaging heat up to 450°F**. But it does more than just feed your fix: it also features **no-stick, light-hold technology** and **delivers frizz control for 72 hours** (or until your next shampoo).



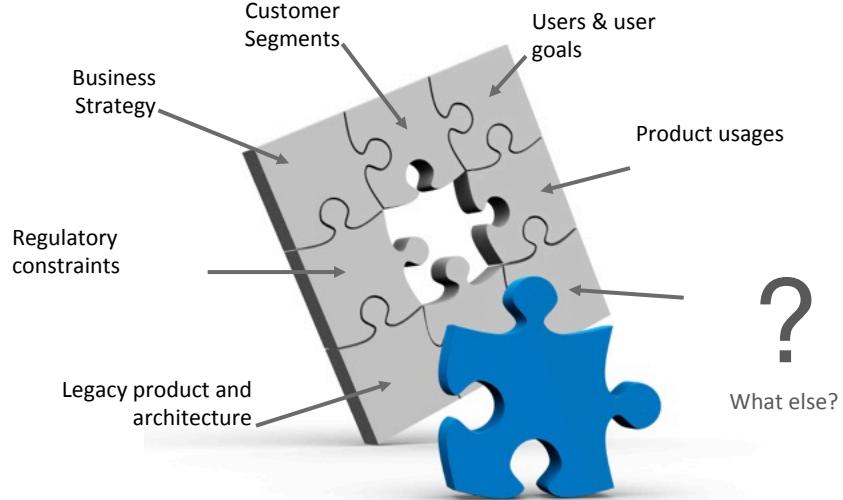
Product Box

- The team makes the assumption that the product will be sold in a shrink-wrapped box,
- Teams design the product box front and back
 - Come up with a product name, a graphic, three to four key bullet points on the front to "sell" the product
 - Include a detailed feature description on the back, and operating requirements.

Jim Highsmith's book [Agile Project Management: Creating Innovative Products](#)



Learn about context to identify the best solution
- involve your team



© 2009 Jeff Patton, all rights reserved, www.AgileProductDesign.com



Create an Initial Product Vision

- Each team will come up with a product idea that you will turn into a product vision and decompose into user stories.
- Using the elevator statement format, discuss your product idea and put together an initial product vision.

15 Minutes



User Roles & Personas



Who are our customers?

Understanding Users

- What **types** of users will use this product?
- What **problems** will this solution solve?
- What **goals** will they hope to achieve?
- What **tasks** will they need to perform?
- Which of these tasks will we **support**?

Source: Jeff Patton. "Personas, Profiles, Actors & Roles. Modeling users to target successful product design"
SD Best Practices Conference and Expo 2007.



User Types – Creating User Profiles

- Construct a profile for each user/role type
- A profile contains relevant information about the user
 1. **# of users** that occupy this user type
 2. **General responsibilities or activities**
 3. **Domain expertise**
 4. **Goals:** how does this product / tool help this user reach their goals?
 5. **Pain Points:** what nagging problems can this product help solve?
 6. **Usage Contexts:** where will this product be used?
 7. **Software Ecosystem:** what other products / tools does this user type rely on?
 8. **Collaborators:** who does this user work with to help reach their goals?
 9. **Frequency of Use:** how often is this type of user likely to use this product?

Source: Jeff Patton. "Personas, Profiles, Actors & Roles. Modeling users to target successful product design"
SD Best Practices Conference and Expo 2007.



Sample profile – The Social Gamer



Steven & Tony

- Early 20's
- Juggles work, friends, girlfriend, playing music and gaming
- Gaming is personal time
- Enjoys playing a good game



Sample profile – The ‘Conqueror’



Stuart

- Mid 20's
- Works in IT
- Passionate about gaming
- Gaming is an escape
- Gaming is an obsession
- Gaming is a secret place
- Plays every non-working, waking moment



Profiles – Get the Facts

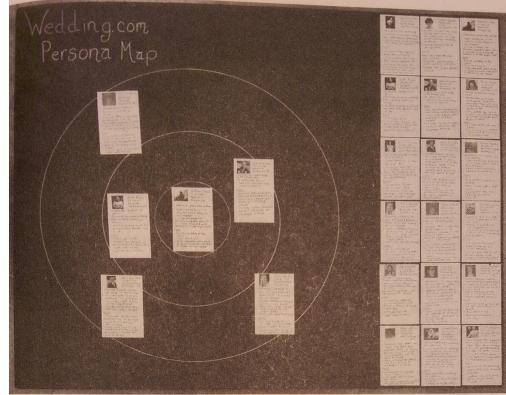
- Identify the areas where your information is thin or incomplete
- Use research to backfill your profiles with facts in critical areas:
 - Interviewing users from target user groups
 - Observing users
 - Questionnaires
 - Existing published demographics
 - Existing published research
 - Customer service records and representatives
 - Sales and marketing
 - Usability testing
 - Focus groups

Source: Jeff Patton. "Personas, Profiles, Actors & Roles. Modeling users to target successful product design"
SD Best Practices Conference and Expo 2007.



Why Create Personas?

- Help with **adding or removing functionality** from scope
- Make **better tactical design decisions**
- Identify user **test subjects**
- Identify **alpha/beta testers**
- Compare with actual users to **validate assumptions**
- **Avoid self-referential design** – We're not building the product for ourselves?



Source: Jeff Patton, "Personas, Profiles, Actors & Roles. Modeling users to target successful product design" SD Best Practices Conference and Expo 2007.

Richard Seridan – Menlo Innovations, Joy, inc.



Pragmatic Persona

Susan – user of the “Nanny App”

▪ Context

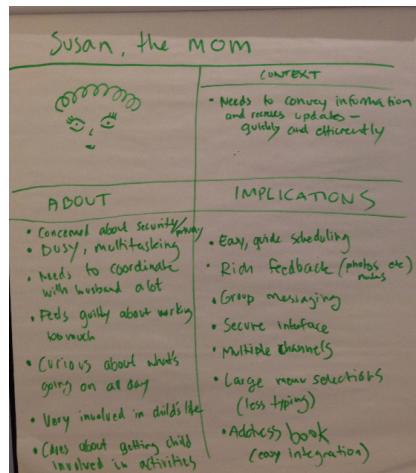
- Needs to communicate and receive information to and from her nanny about her child during the day

▪ About Susan

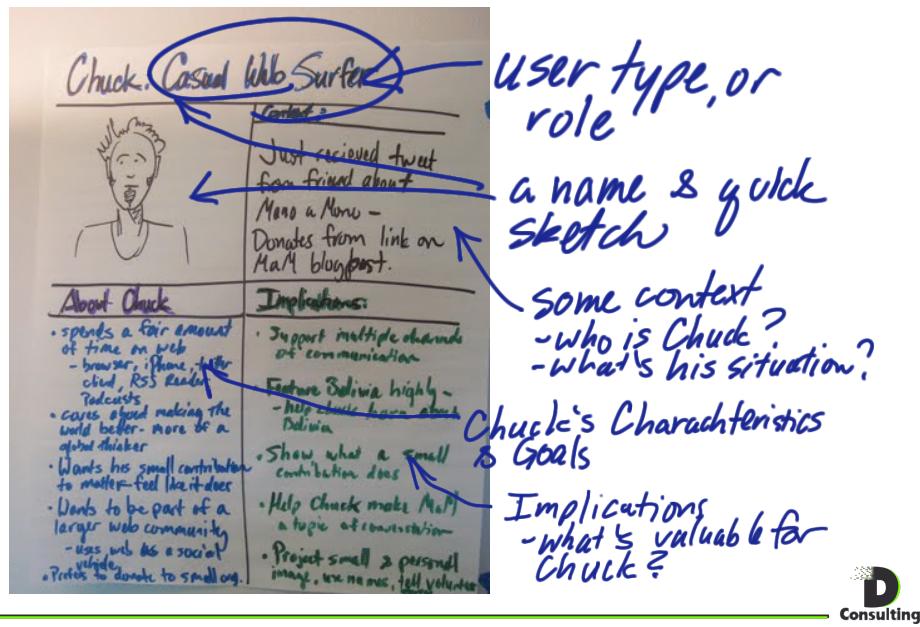
- Working mother – often in meetings making it difficult to answer the phone
- Busy at work, constantly multitasking
- Needs to coordinate child activities with her husband and nanny
- Feels guilty about working and not being home
- Curious about what is going on with her child during the day
- Very involved in the child's life

▪ Implications

- Want something easy to use
- Wants rich feedback – scheduling, photos, texts, alerts
- Group messaging
- Most access will be via smart phone
- Wants rich menu selections to reduce typing
- Would like integration with existing address books



Pragmatic Persona



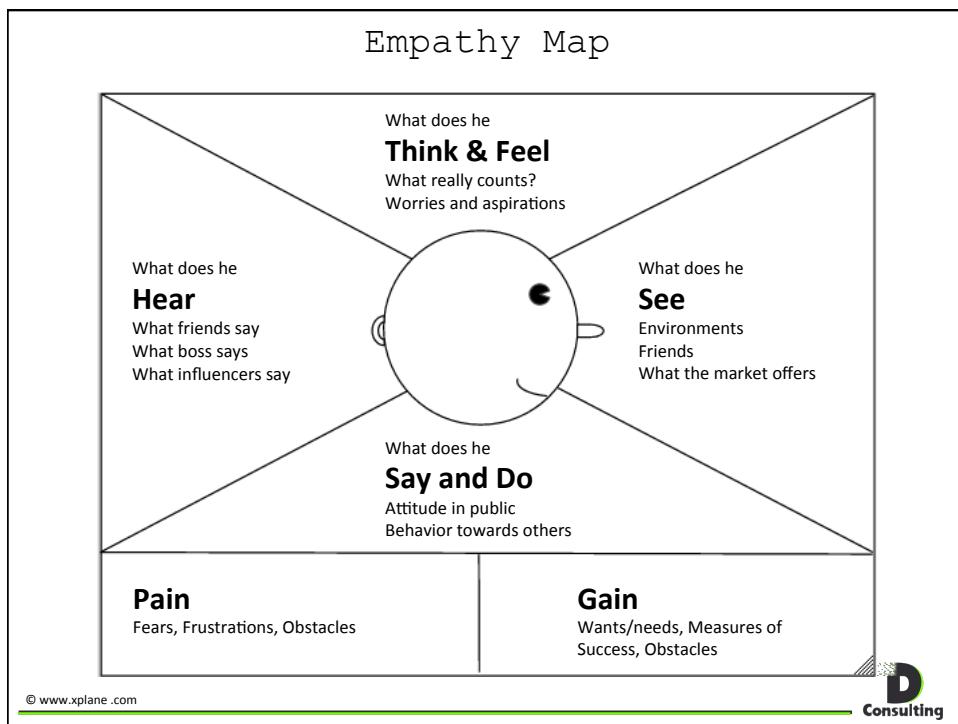
user type, or role

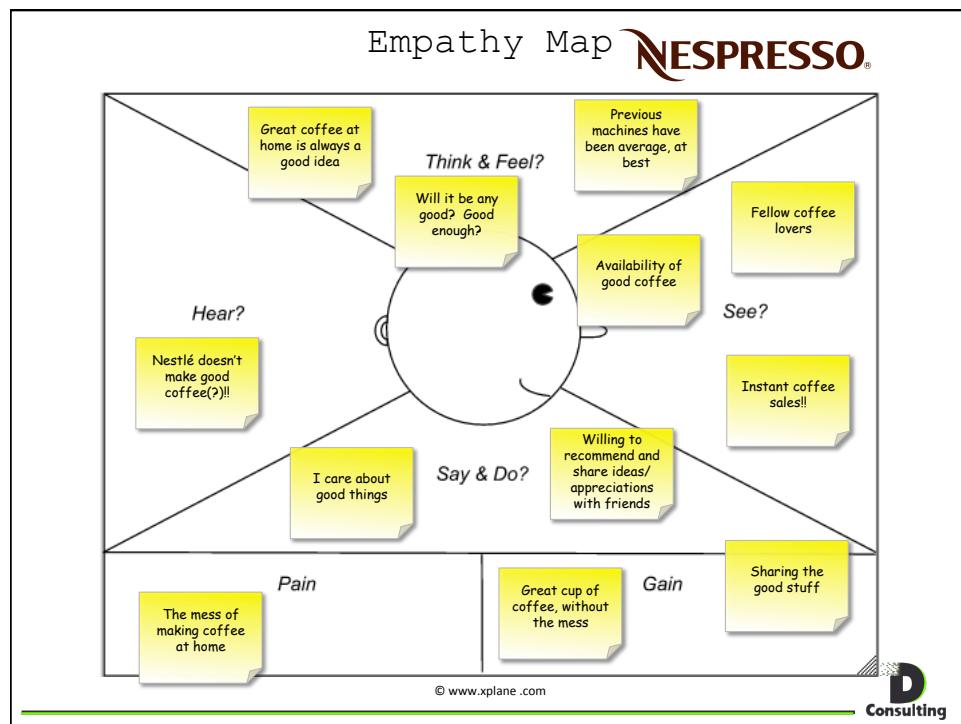
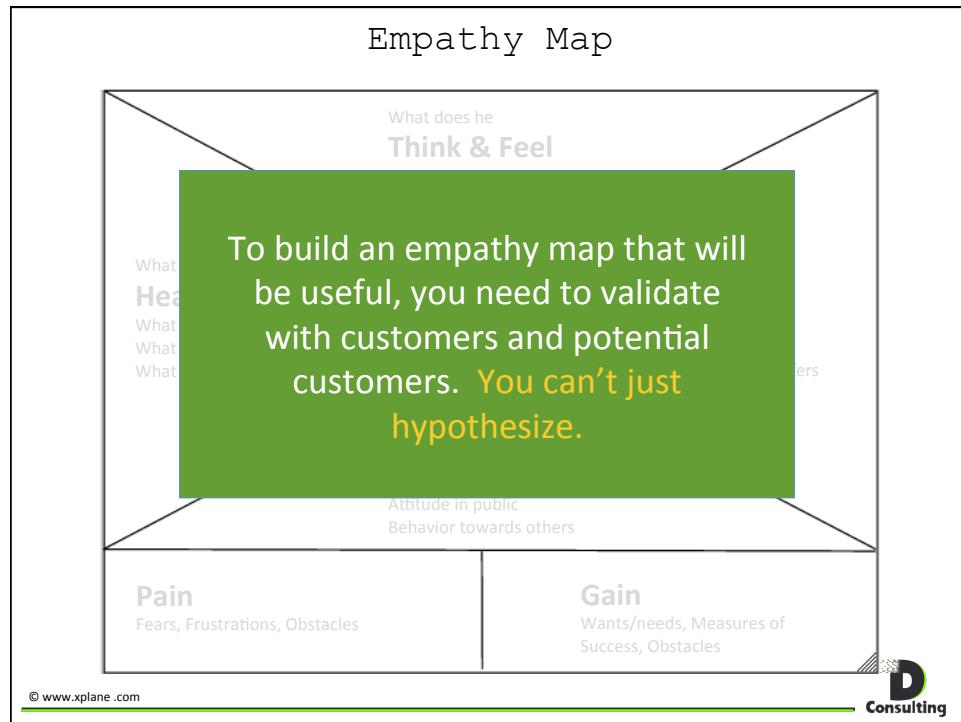
a name & quick sketch

some context
-Who is Chuck?
-What's his situation?

Chuck's Characteristics & Goals

Implications
-What's valuable for Chuck?





User role exercise

- Identify two users of your product, use an empathy map for one and pragmatic persona for the other.

25 minutes

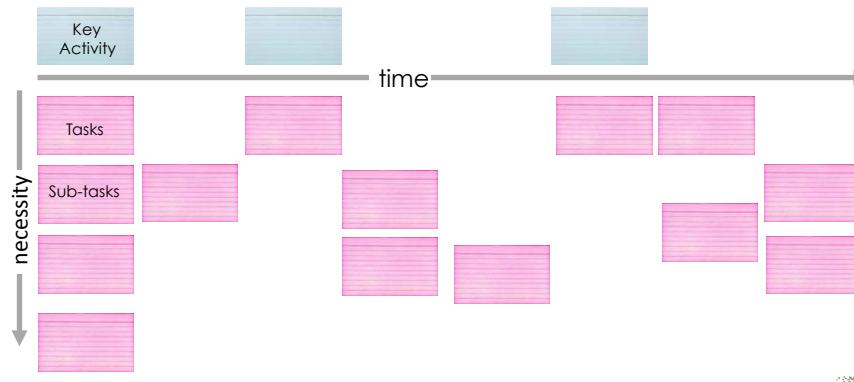


User Experience Mapping



User Experience or Story Mapping

- Above the line - arrange user activities, features or epics left to right in the order you'd explain them to someone when asked the question: *"What do people do with this system?"*
- **Add themes or task-centric stories** in under each activity in chronological order left to right.
 - If you were to explain to someone what a person typically does in this activity, arrange tasks in the order you'd tell the story
- **Overlap user tasks vertically** if a user may do one of several tasks at approximately the same time
 - If in telling the story I say the systems' user typically "does this or this or this, and then does that," "or's" signal a stacking vertically, "and then's" signal stepping horizontally.

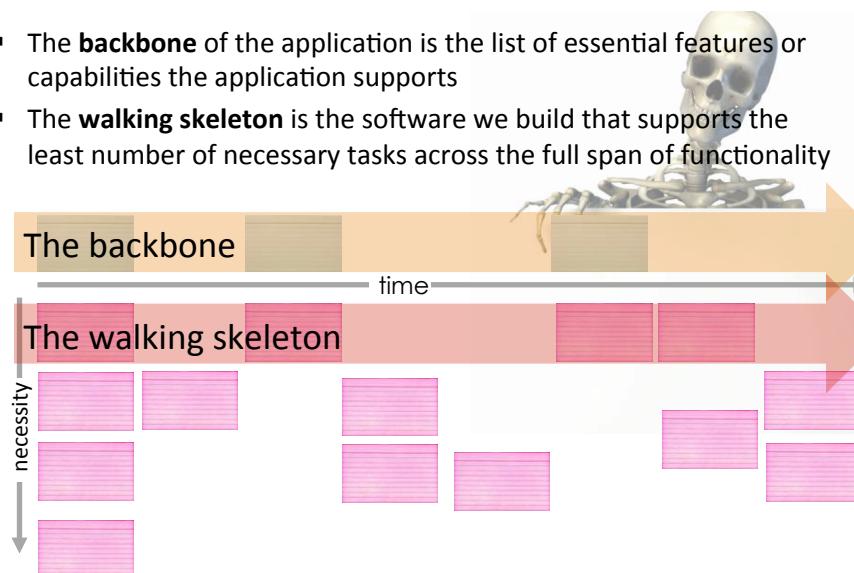


Source: Jeff Patton. "An Introduction to User Story Mapping"



Two important “anatomical” features

- The **backbone** of the application is the list of essential features or capabilities the application supports
- The **walking skeleton** is the software we build that supports the least number of necessary tasks across the full span of functionality

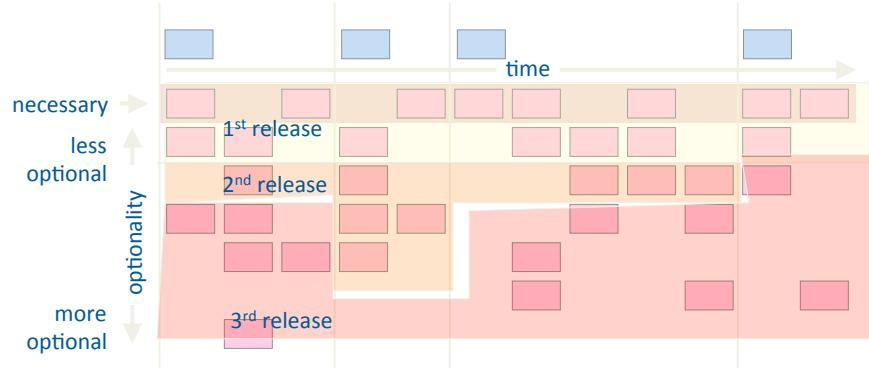


Source: Jeff Patton. "An Introduction to User Story Mapping"



Identify releases in a Map by slicing horizontally

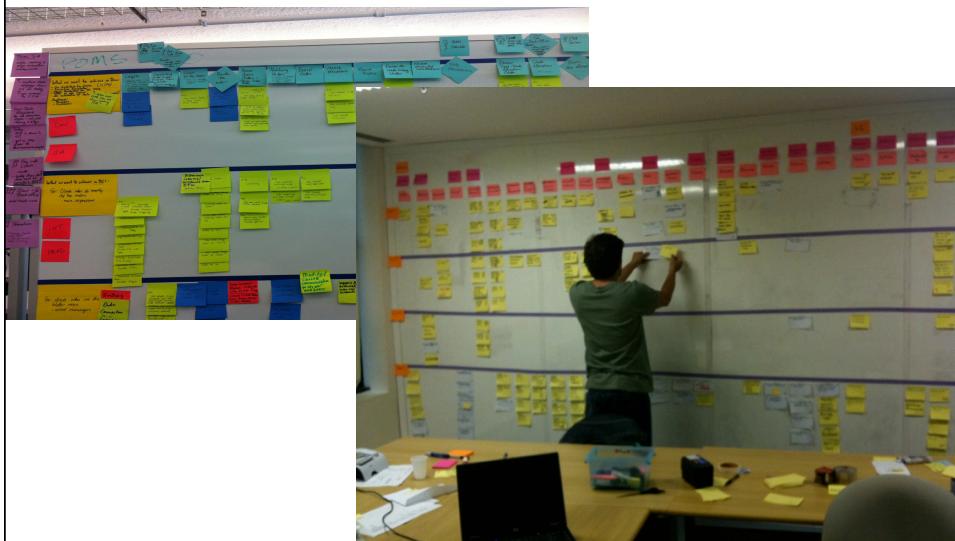
- Choose **coherent groups of features** that consider the span of business functionality and user activities.
- Support all necessary user activities with the first release
- Improve activity support with subsequent releases



Source: Jeff Patton. "An Introduction to User Story Mapping"



Example Story Map



For more information on user story/experience mapping – see Jeff Patton's presentation:
http://www.agileproductdesign.com/blog/the_new_backlog.html



Exercise

- From your product vision create an initial User Experience Map in your teams

25 mins



Release Planning

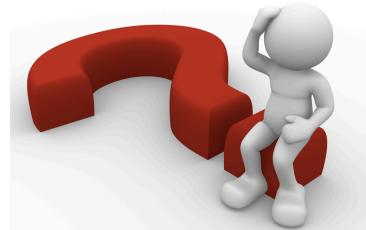


Release planning questions

- How much will be done by...?
- When can we ship with this set of features?
- How much will it cost to get this set of features?

Inputs

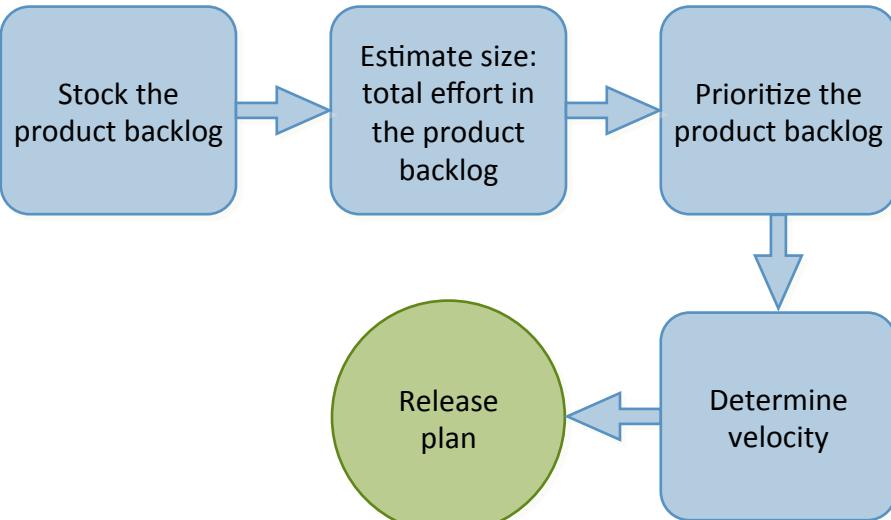
- Velocity - The amount of work completed in a sprint
- Prioritized product backlog
- Estimates



Mountain Goat Software – Mike Cohn <http://www.mountaingoatsoftware.com/>

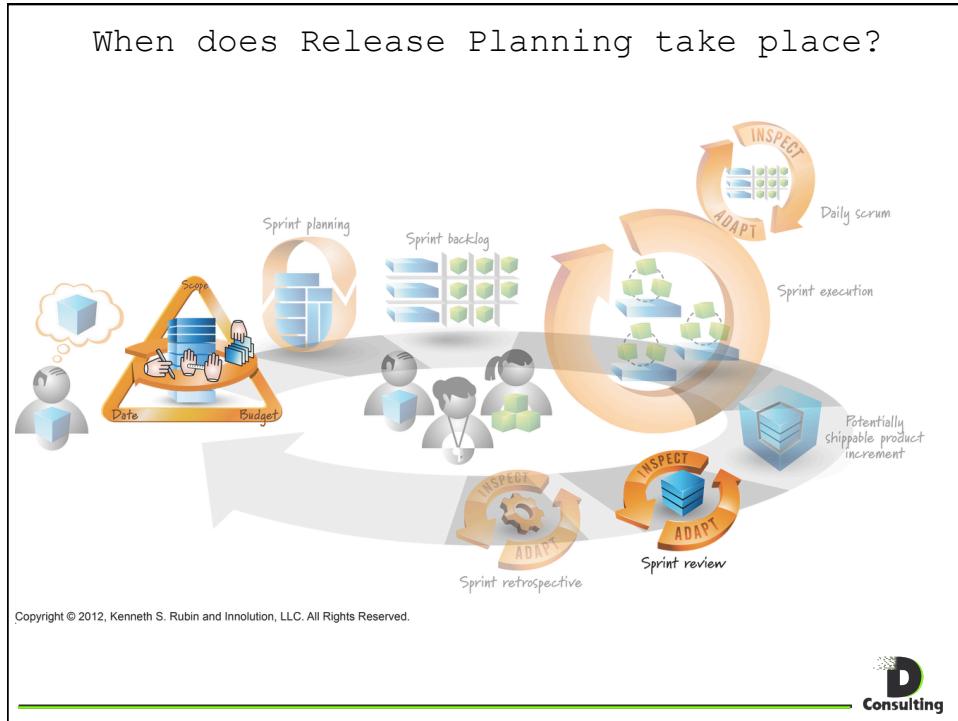
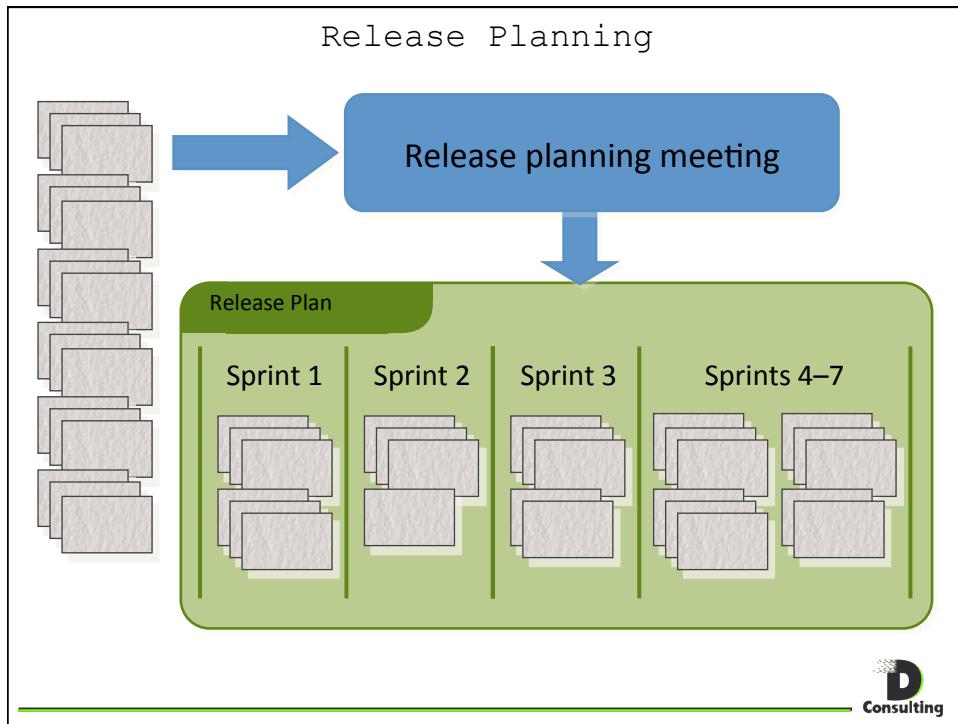


Release Planning Process



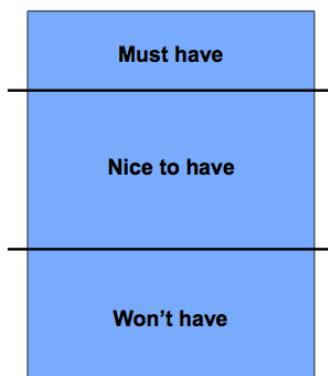
Mountain Goat Software – Mike Cohn <http://www.mountaingoatsoftware.com/>





What features to include ?

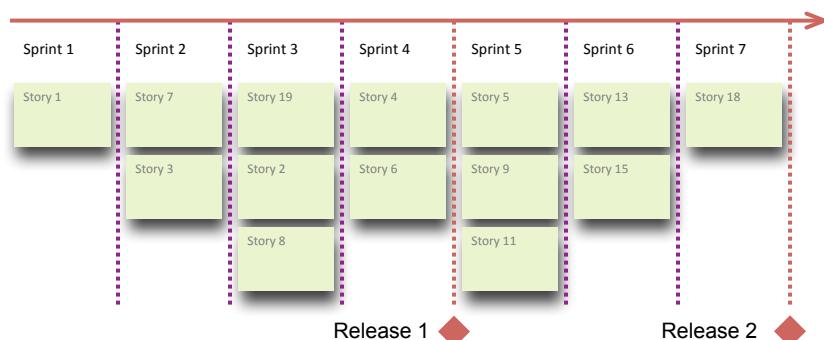
Product Backlog



- Have to ship with these features – can't ship without them
- Would like to get these features into the release, but will ship without them
- Moved to the next release (out of scope)



Project Layout



- Projects comprise of **multiple releases**
- Each release comprises of **multiple sprints**
- Stories are **started and completed** within each sprint
- Stories are executed based on **value and priority**



Always accurate, sometimes precise

- Plans must always be accurate
- BUT gain precision over time
 - “We’ll be done between September and November.”
 - “We’ll be done sometime in November.”
 - “We’ll be done on November 7.”
- How precisely can you estimate next years plan?
 - How precisely do you need to?

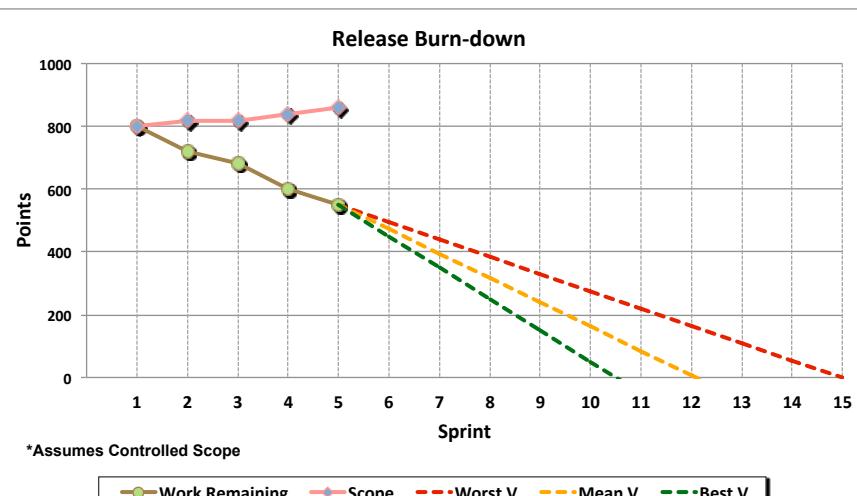
“It’s better to be roughly right than precisely wrong”

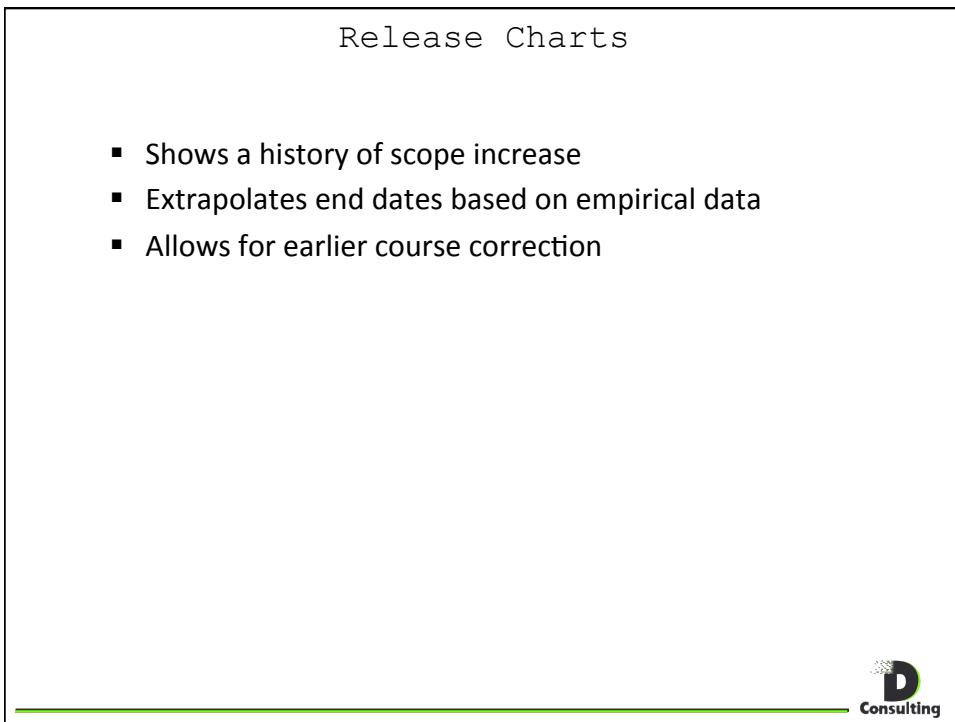
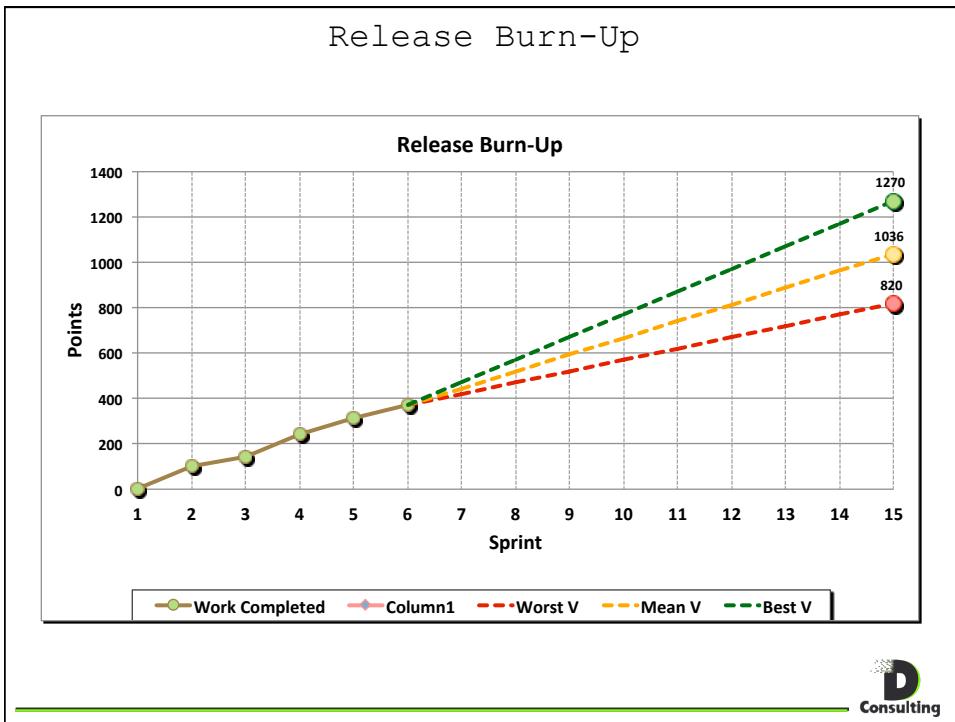
~John Maynard Keynes

Mountain Goat Software – Mike Cohn <http://www.mountaingoatsoftware.com/>



Release Burn-Down – Multiple Velocities





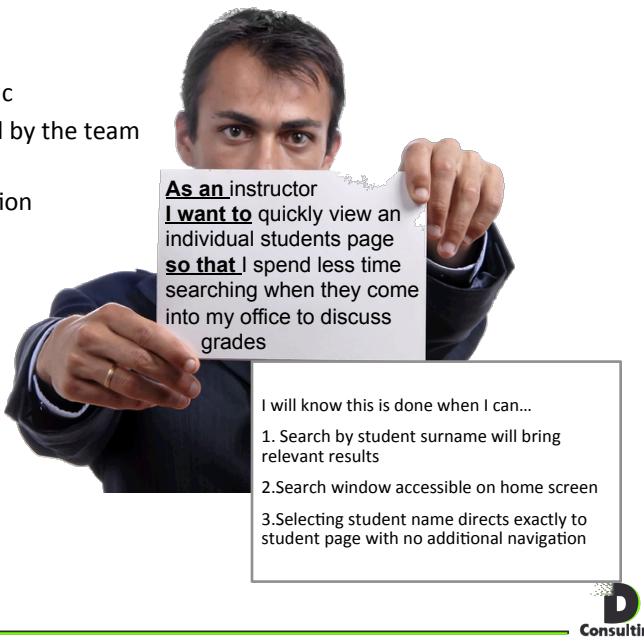
The User Story: The Fundamental Unit of Work in Agile

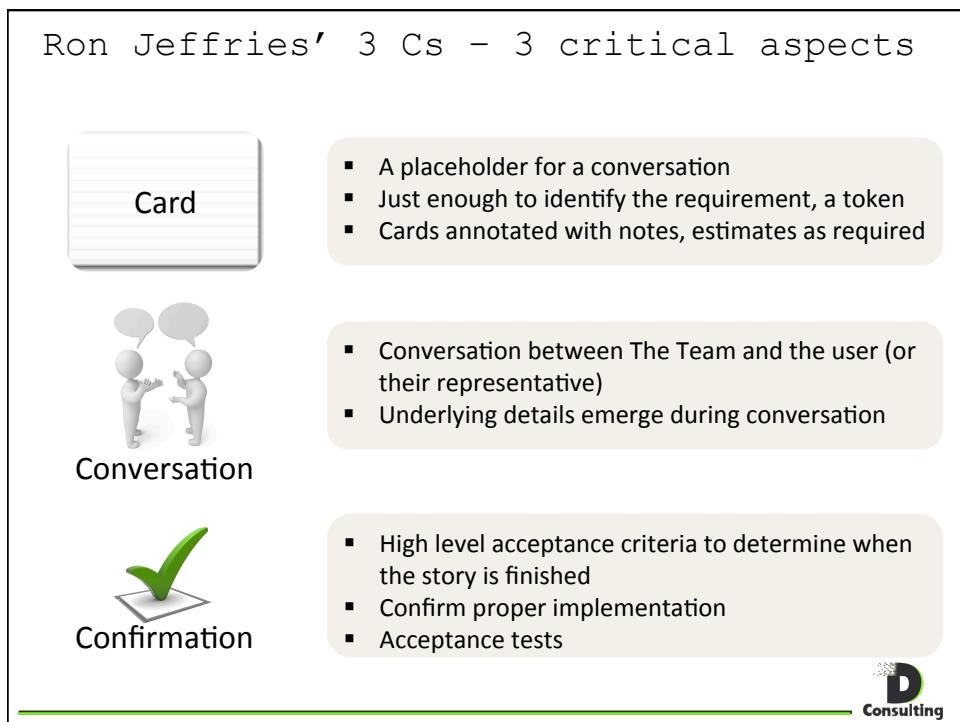
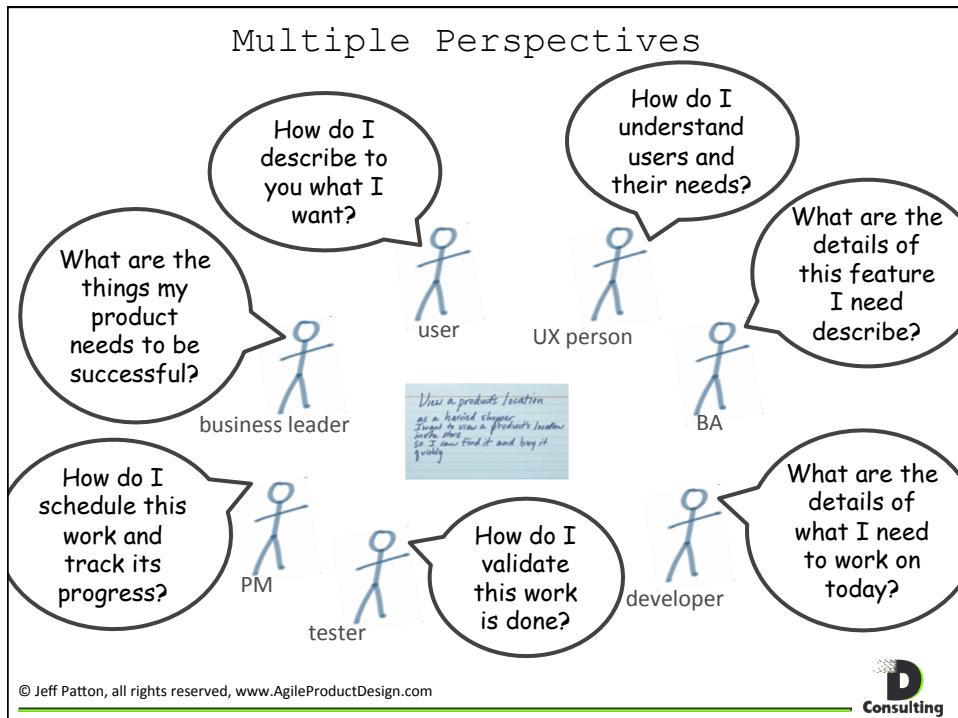


Anatomy of a user story

Best Practices:

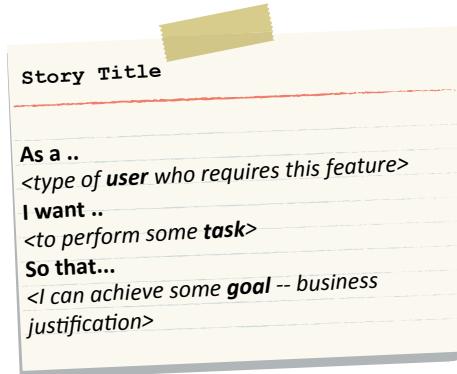
- Technically agnostic
- Can be understood by the team and the customer
- Contain a justification



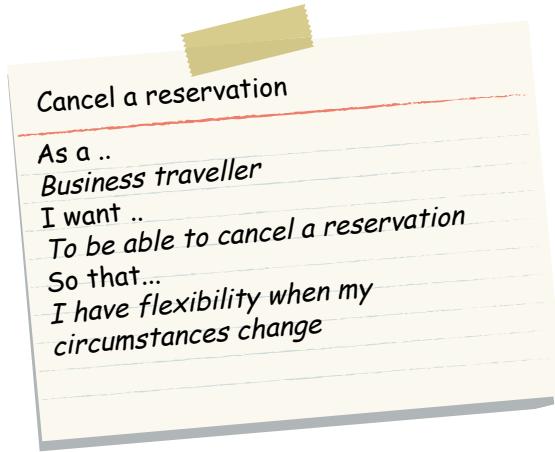


A Template for User Stories

- Written from a user/business perspective to emphasize user goals, not system attributes. Usually expressed as follows:



Example Story - A card



Example Story - A confirmation

I'll know when this is done when;

- Verify that a premium member can cancel the same day without a fee
- Verify that a non-premium member is charged 10% for a same-day cancellation
- Verify the email confirmation is sent
- Verify that the hotel is notified of any cancellation



What makes for a good story?

Follow the I.N.V.E.S.T. principle

independent

- Avoids dependencies
- Combine stories or split if there are dependent elements

negotiable

- Stories are not written contracts
- Flexible about implementation

valuable

- Should be written to emphasize business value

estimable

- Stories are elements of planning and have to be estimable (high level)
- need to be understood by the development team for estimation purpose

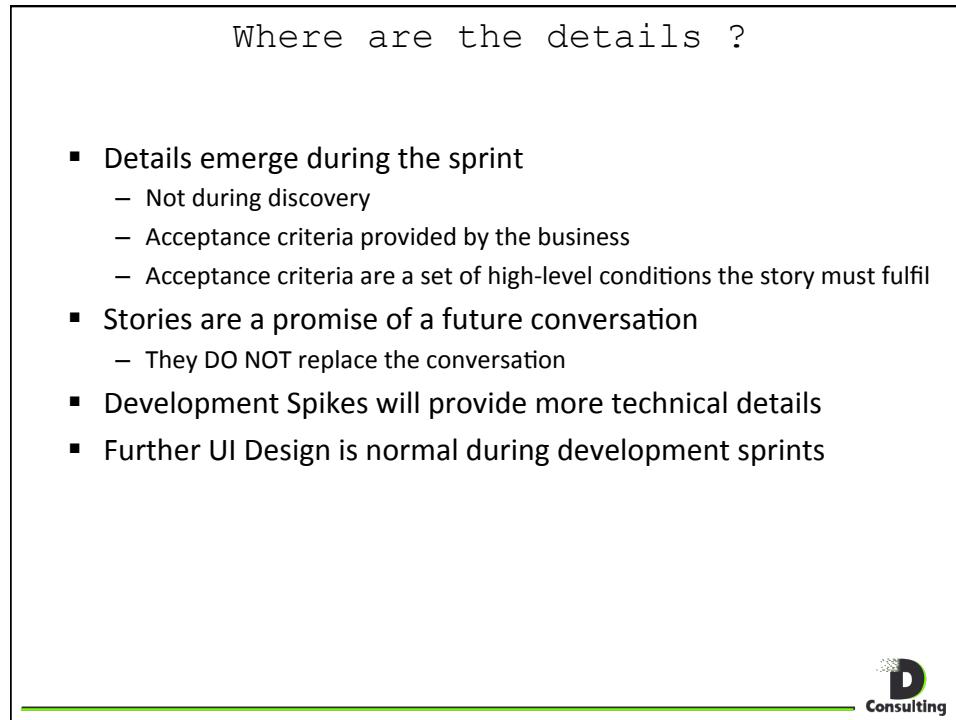
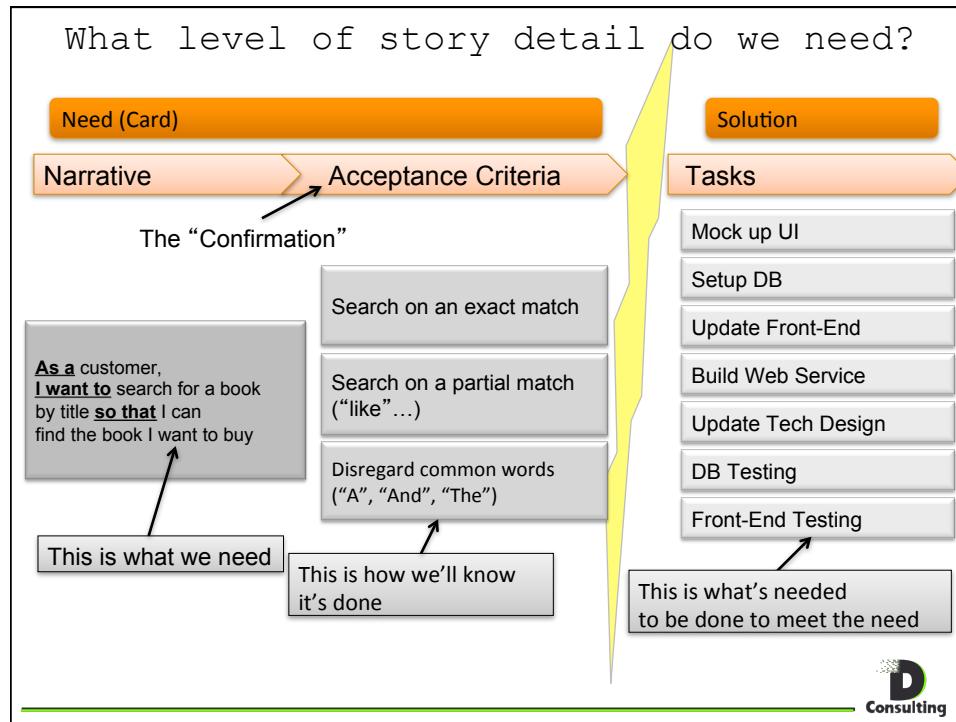
small

- Small stories are easier to estimate and plan
- Large stories don't fit well into time-boxes
- Must be executable within single sprint

testable

- Completeness can be objectively assessed
- Should not be vague





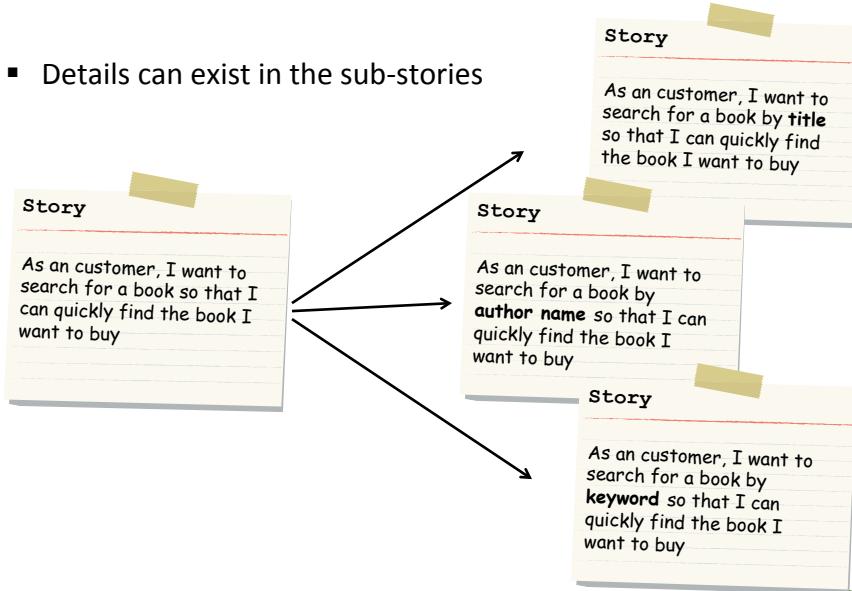
Details as acceptance criteria

- Details can be written as conditions of satisfaction



Details as smaller sub-stories

- Details can exist in the sub-stories



Given, When, Then...

As a customer,
I want to withdraw cash from an ATM,
so that I don't have to wait in line at the bank

Scenario 1: Account is in credit

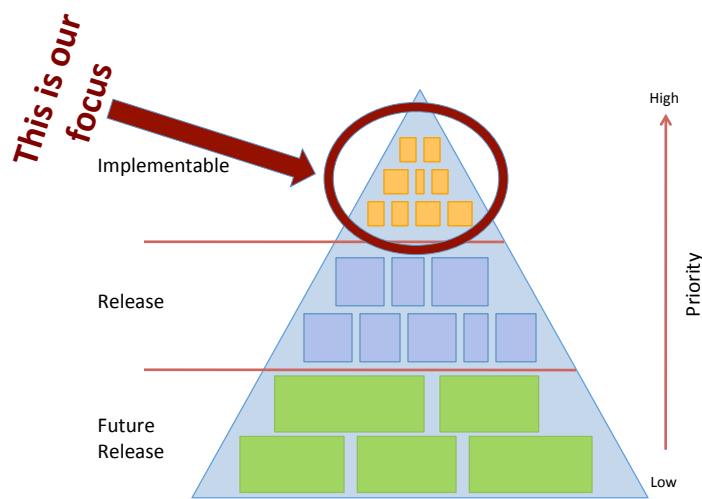
Given the account is in credit
And the card is valid
And the dispenser contains cash
When the customer requests cash
Then ensure the account is debited
And ensure cash is dispensed
And ensure the card is returned

Scenario 2: Account is overdrawn past the overdraft

limit
Given the account is overdrawn
And the card is valid
When the customer requests cash
Then ensure a rejection message is displayed
And ensure cash is not dispensed
And ensure the card is returned



The Product Backlog Iceberg

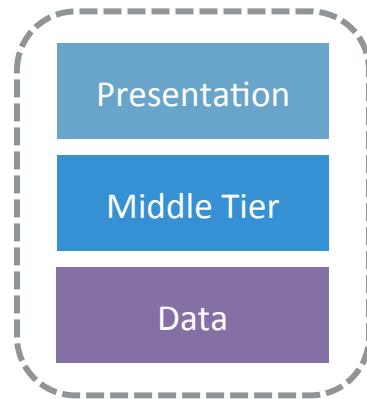


Source: Mike Cohn, Mountain Goat Software

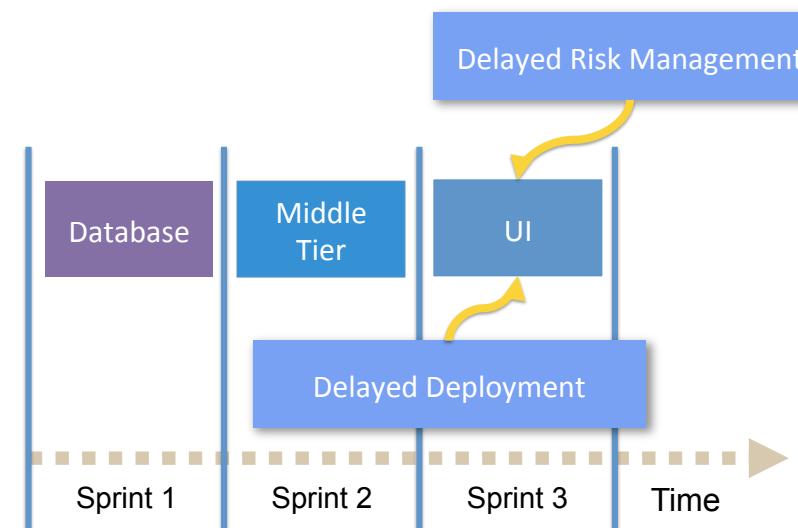


Strive for a Vertically Integrated Story

- All parts of the story are built together in a single Sprint
- Provides fast feedback
- Supports early deployment of functionality – if desired

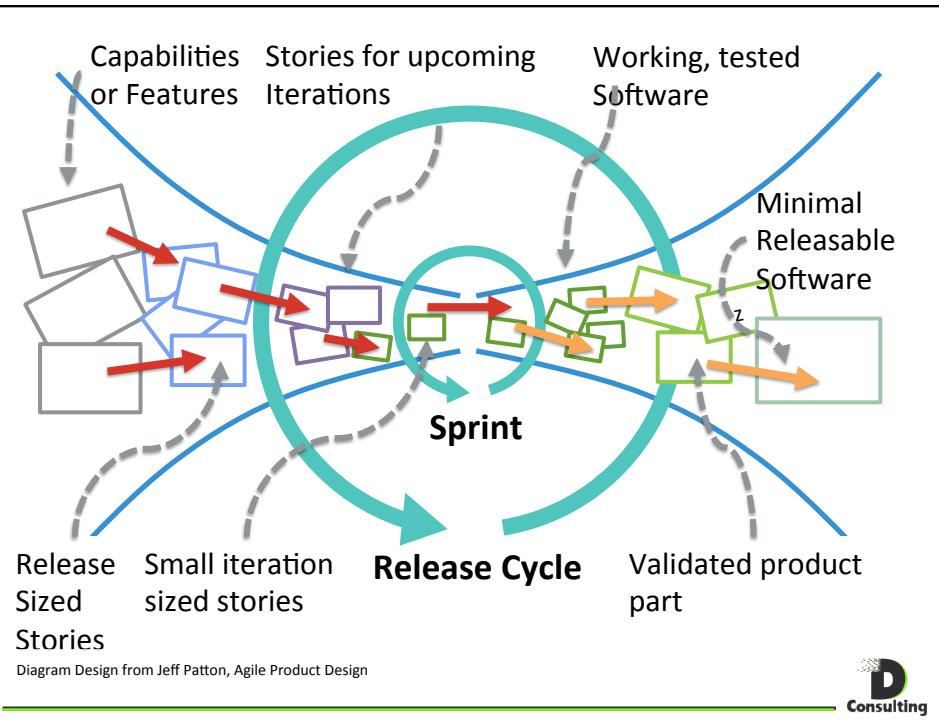


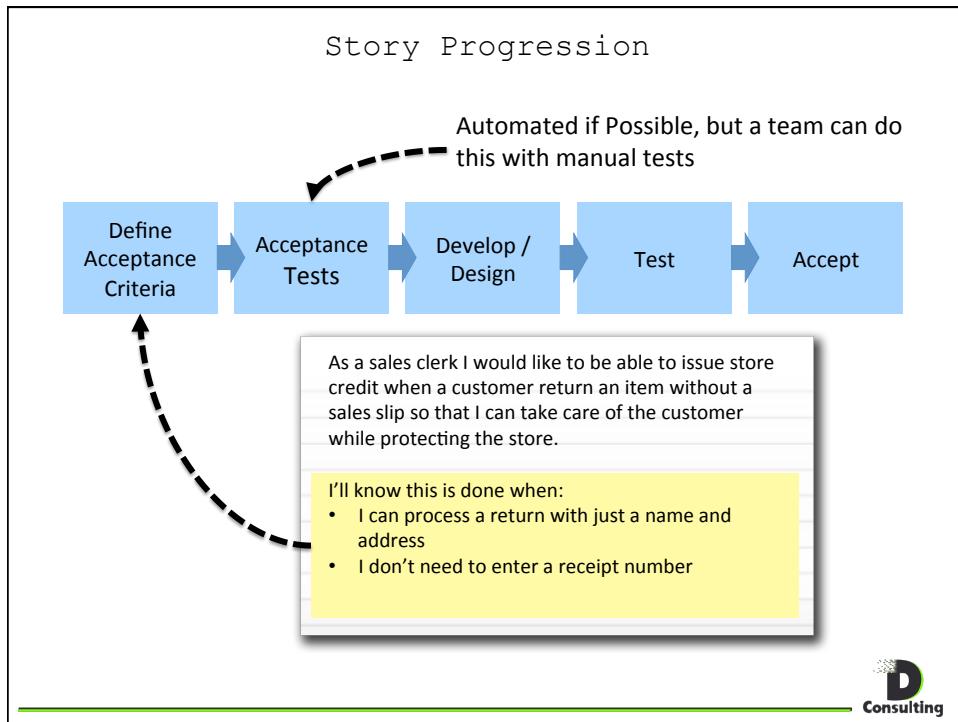
The Challenge of a Horizontal Approach



If you can't complete a unit of potentially-vertically integrated software in an iteration,
then you can't.

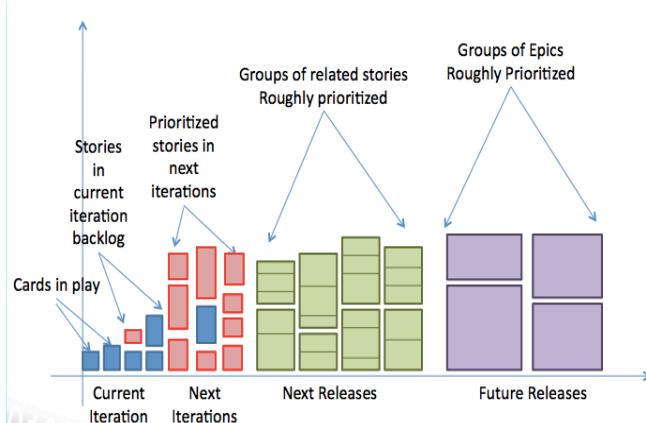
Preferably that is the exception,
not the rule.





Breaking Down Stories

Don't break down too soon - progressively elaborate



Breaking Down or Decomposing Stories

- Agree on target story size with the development team
- If the story is deemed as too big it should be “broken down” or decomposed into smaller, independent stories
- Breaking down stories is a joint effort with customer and development team
- A good indicator that a story is too big can be any conjunctions in your story “As a User I need to be able to do x and y or z.....”
- These smaller stories can still be “grouped” under an appropriate Theme or Epic



INVEST in good user stories

Stories should be:

I – Independent
N – Negotiable
V – valuable
E – Estimable
S – Small
T – Testable



Splitting Stories

Split stories along dimensions of functional complexity

- Workflow Steps
- Business Rule Variations
- Major Effort
- Simple/Complex Cases
- Variations in Data
- Data Entry Methods
- Operations (i.e. CRUD)
- Defer Performance
- Break Out a Spike

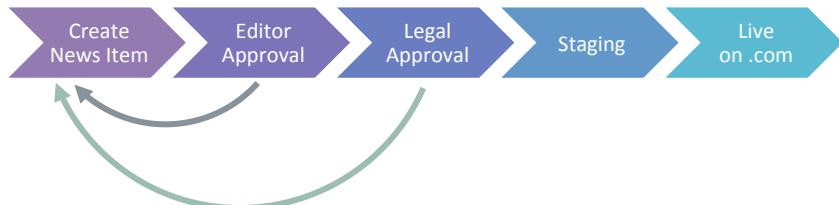


Workflow Steps

As a content manager, I
can publish a news story
to the corporate website.



Workflow Steps



Workflow Steps

As a content manager, I can publish a news story to the corporate website.

- ...I can publish a news story directly to the corporate website.
- ...I can publish a news story with editor review.
- ...I can publish a news story with legal review.



Business Rule Variations

As a user, I can search for flights with flexible dates.

- ...as “n days between x and y.”
- ...as “a weekend in December.”
- ...as “± n days of x and y.”



Major Effort

As a user, I can pay for my flight with VISA, MasterCard, American Express, and other...

...I can pay with one credit card type (of VISA, MC, AMEX, other).

...I can pay with all four credit card types (VISA, MC, AMEX etc).



Simple / Complex

As a user, I can search for flights between two destinations.

...specifying a max number of stops.

...including nearby airports.

...using flexible dates.

...etc.



Variations in data

As a content manager, I
can create news stories.

...in English.
...in Japanese.
...in Arabic.
...etc.



Data entry methods

As a user, I can search
for flights between two
destinations.

...using simple date input.
...with a fancy calendar UI.



Defer performance

As a user, I can search for flights between two destinations.

... (slow - just get it done, show a “searching” animation).
... (in under 5 seconds).



Operations

As a user, I can manage my account.

... I can sign up for an account.
... I can edit my account settings.
... I can cancel my account.



Break out a spike

As a user, I can pay by credit card.

Investigate credit card processing.
Implement credit card processing.



What about non-functional Requirements?

As acceptance criteria

As a customer, I want to be able to pay by PayPal, so that I can complete my purchase

- payment confirmed within 5 seconds
- handle 100 concurrent payments
- encrypted redirect to PayPal

As separate user stories

As a user, I want the site to be available 99% of the time I try to access it, so that I don't get frustrated and find another site to use.

As a user, I want an interface in English, a Romance language and a complex language.

As constraint cards

As the CTO, I want the system to use our existing orders database rather than create a new one, so that we don't have one more database to maintain

As delivery process

Definition of done:

- deployed to test environment
- dev box review with PO
- architectural documentation updated
- StyleCop analysis no warnings
- handover to test completed

Prioritization



Approaches to Prioritization

- Expert opinion
- Priority poker
- Buy a Feature
- Theme screening
- Theme scoring
- Relative Value Scoring
- Kano Analysis
- Risk
- Minimal Viable Product



Expert Opinion

- Focus on delivering value to customer
- But consider these four factors
 - Delivery of new capabilities
 - Development of new knowledge
 - Mitigation of risk
 - Changes in relative cost



Priority Poker

- Place a card for each theme on the table
 - Put individual stories underneath the theme
- On each theme, stack poker chips equal to the estimated cost
- Figure out how much you can “buy” during the release period
 - e.g., 12 sprints × velocity of 20 = 240
- Product owner(s) take that many fake bills or other colored poker chips
 - Buy themes by placing bills/chips on the themes



Priority Poker



Buy A Feature – Innovation Games ®

- A list of 12-20 items (features or projects) are described in terms of benefits and cost
- 5 to 8 invited stakeholders given limited “budget”, must reach consensus on projects to “buy”
- Captures very rich information about customer motivations, trade-offs, objections, actual collective needs

In-person

- Provides rich opportunity for “new” ideas

Or Online

- Captures data for greater analysis of preferences
- Preliminary trials indicate faster/more accurate results than traditional tools



Theme Screening

- Identify 5-9 (approximately) selection criteria for what is important in the next release
- Select a baseline theme
 - Likely to be included in the next release
 - Understood by most team members
- Assess each candidate theme relative to the baseline theme



Theme Screening

Selection Criteria	Themes						
	Theme A	Theme B	Theme C	Baseline Theme	Theme E	Theme F	Theme G
Importance to existing customers	+	+	-	0	-	+	0
Competitiveness with ABC Corp.	+	-	0	0	0	0	0
Starts us integrating product lines	+	0	0	0	+	+	+
Generates revenue in Q2	0	0	0	0	+	0	+
+ = better than	Net score		3	0	-1	0	1
0 = same as	Rank		1	4	5	4	3
- = worse than	Continue?		Y	N	N	Y	Y



Theme Scoring

- Like theme screening but selection criteria are weighted
- Need to select a baseline theme for each criteria
 - Avoids compression of a category
- Each theme is assessed against the baseline for each selection criteria

Much worse than reference	1
Worse than reference	2
Same as reference	3
Better than reference	4
Much better than reference	5



Theme Scoring example

Selection Criteria	Weight	Theme A		Theme B		Theme C	
		Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
Importance to existing cust.	25	3	0.75	1	0.25	4	1.00
Competitive. with ABC	10	2	0.20	3	0.30	3	0.30
Starts us integrating...	15	3	0.45	4	0.60	4	0.60
Generates Q2 revenue	50	5	2.50	2	1.00	3	1.50
	Net score		3.90		2.15		3.40
	Rank		1		3		2
	Continue?		Yes		No		Yes



Relative Value Scoring

1. Rate the **positive impact of implementing** a feature, epic or theme from 1-9
2. Assess **negative impact of NOT implementing** it from 1-9
3. Calculate the value of each theme relative to the entire product backlog to get the **relative value** of that theme
4. Feature size is any **sum of story sizes**
5. Calculate cost of each theme relative to the entire product backlog
6. Priority is calculated by ($\text{Relative Value} \div \text{Relative Cost}$)
7. Particularly useful for assessing priority beyond the “walking skeleton”



Relative Value Scoring

Feature	Relative Benefit	Relative Penalty	Total Value	Value Percent	Estimate	Cost Percent	Priority
Feature A	8	6	14	30	40	24	123 2
Feature B	7	5	12	26	30	18	140 1
Feature C	9	2	11	23	50	30	77 4
Feature D	1	9	10	21	45	27	78 3
Total			47	100	165	100	

Total Value = Relative Benefit + Relative Penalty

Value Percent = Total Value / $\Sigma(\text{Total Value})$

Cost Percent = Estimate / $\Sigma(\text{Estimate})$



Kano Analysis

- First described by Dr. Noriaki Kano
- Describes the **complexities of customer needs** and their relationship to customer satisfaction in a visual format
- Provides insight into **attributes perceived as important** to customers. Helps teams focus on differentiating features.
- Divide themes into **Exciter** ("Wow! I didn't know you could do that!") , **Linear** ("The more, the better") and **Mandatory** ("Must have") features.
- Implement the **Mandatory** themes
 - Absence leads to dissatisfaction
 - Presence contributes to customer neutrality
- Add as many **Linear** (Performance) themes as possible
- Add some **Exciters** (Delighters) to wow the customers



Kano Criteria

Three types of features

Mandatory /
Baseline / Basic

Must be present in order for
users to be satisfied

Linear /
Performance

The more, the better

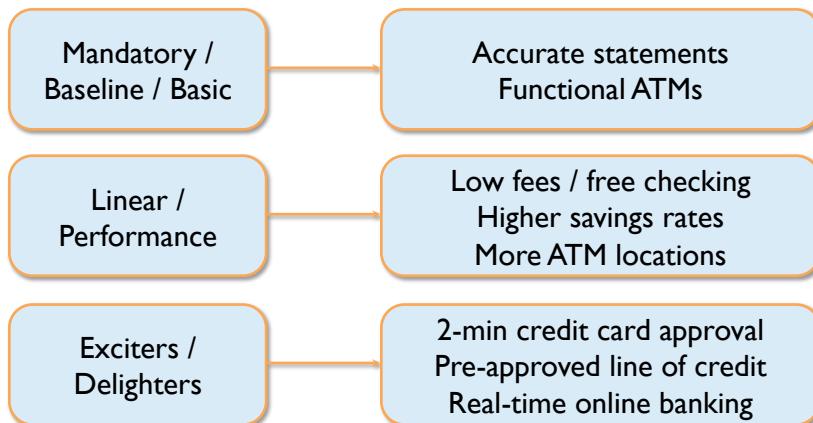
Exciters /
Delighters

Functionality a user doesn't know
he wants, until he sees it

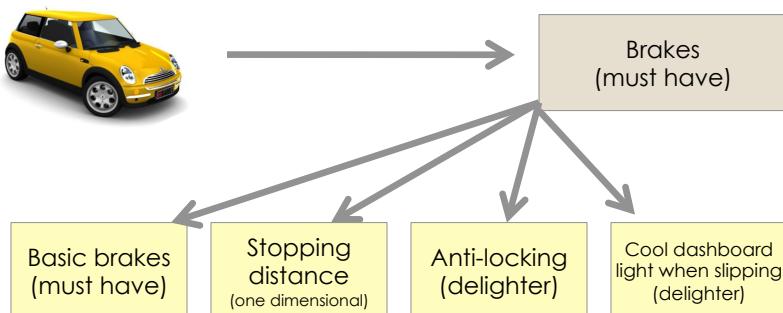


Kano Analysis Example

Criteria for Banking Features



Prioritize and Split



Keep in mind: you must know your customers and users to determine subjective value.

One person's delighter may leave others apathetic.

Another's must have is useless to other customers

Risk & Prioritization

Anything that might happen that would jeopardize or limit the success of the project.

Types of Project Risk:

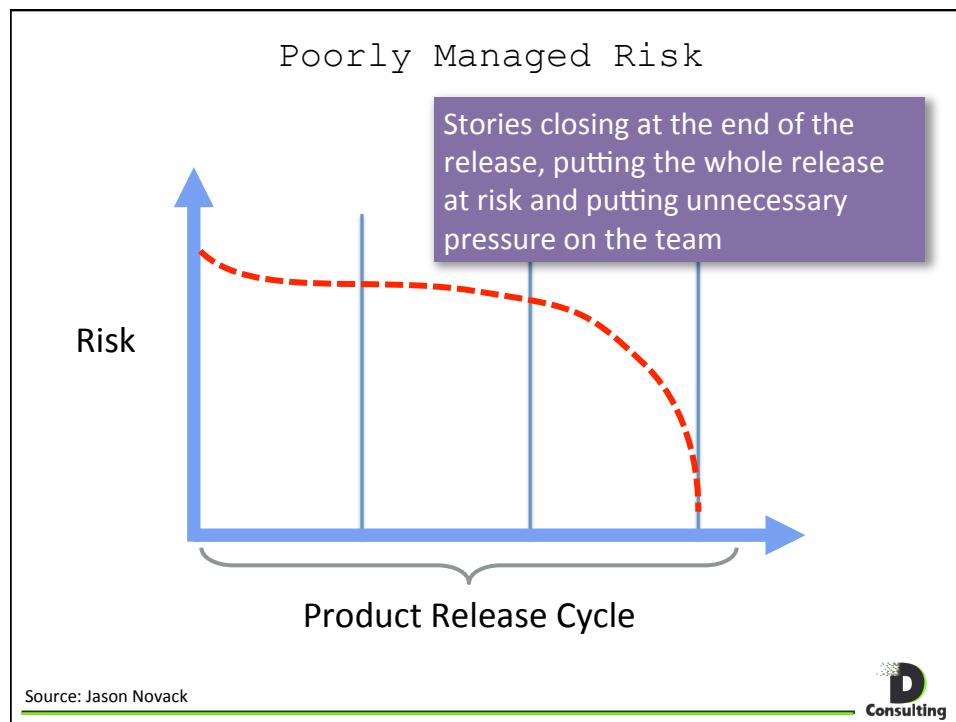
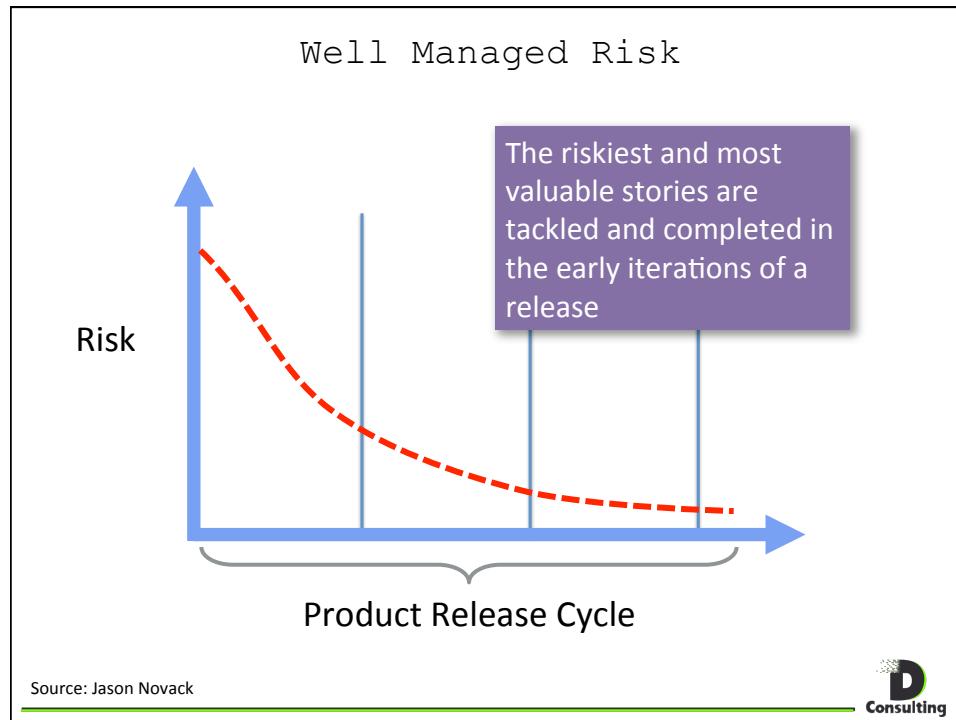
- Schedule Risk
 - “We may not be able to complete this feature in time”
- Cost Risk
 - “This feature may cost more than we expect”
- Functionality Risk
 - “This workflow may be all wrong”
- Technical Risk
 - “This may not perform at the level we need it to”



Risk Value Prioritization

Which stories should we work on first?





MVP – Minimal Viable Product

- Minimum Viable Product is the smallest batch you can do to learn about your hypothesis from your customers
- Minimum Viable Product is not the same as a Minimum Marketable Product
- Minimum Viable Product does not always mean code



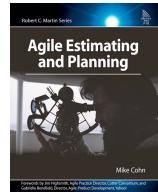
Closing Activities

- Parking lot
- Recommended reading list and discussion groups

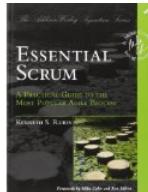


Recommended Reading

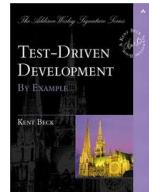
Agile Estimating and Planning
By Mike Cohn



Essential Scrum
By Kenneth S. Rubin



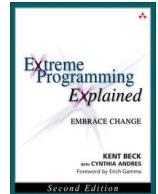
Test-Driven Development
By Example
By Kent Beck



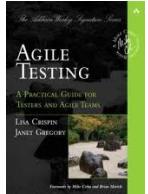
User Stories Applied
By Mike Cohn



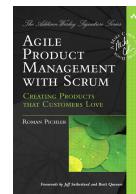
Extreme Programming Explained
By Kent Beck



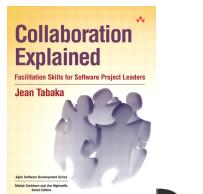
Agile Testing
By L. Crispin, J. Gregory



Agile Product Management with Scrum
By Roman Pichler



Collaboration Explained
By Jean Tabaka



Other Mentioned and Notable Books

- *Specification by Example*, Gojko Adzik, 2011.
- *Scrum and XP from the Trenches*, Henrik Kniberg, 2007.
- *Joy, Inc.* Richard Sheridan, Portfolio Hardcover, 2013.
- *Crossing the Chasm*, Geoffrey A. Moore, 2006-2014.
- *The Entrepreneurs Guide to Customer Development*, B. Cooper & P. Vlaskovits, 2010.
- *Fearless Change: Patterns for Introducing New Ideas*, Linda Rising, 2004.
- *Innovation Games*, Luke Hohmann, 2007.
- *Business Model Generation*, Alexander Osterwalder, Yves. Pigneur, 2010
- *Lean Software Development*, Poppendieck and Poppendieck, Addison Wesley, 2003.
- *Agile Project Management with Scrum*, Ken Schwaber, Microsoft Press, 2004.
- *Agile Retrospectives: Making Good Teams Great*, Esther Derby and Diana Larsen, Pragmatic Bookshelf, 2006.
- *Agile Software Development with Scrum*, Ken Schwaber and Mike Beedle, Prentice Hall, 2002.
- *Project Retrospectives*, Norman Kerth, Dorset House, 2001.
- *The Enterprise and Scrum*, Ken Schwaber, Microsoft Press, 2007.



Other Resources

- Scrum Gatherings – see Scrum Alliance
- Agile Conference – see Agile Alliance
- Find Agile Meetups at <http://www.meetup.com/>
- Scrum Guide <http://www.Scrum.org>
- Scrum Alliance <http://www.scrumalliance.org>
- Agile Alliance <http://www.agilealliance.org>
- Agile Manifesto <http://www.agilemanifesto.org>
- Mike Cohn <http://www.mountaingoatsoftware.com>
- Jeff Patton <http://www.agileproductdesign.com>
- Jeff Sutherland <http://www.JeffSutherland.com>



- One or more slides in this presentation contain items from the Visual AGILExicon™, which is a trademark of Innolution, LLC and Kenneth S. Rubin.
- The Visual AGILExicon is used and described in the book: ***Essential Scrum: A Practical Guide to the Most Popular Agile Process***.
- You can learn more about the Visual AGILExicon and permitted uses at: <http://innolution.com/resources/va-home-page>

