

Grundlegende Begriffe (K1)

System („ein aus mehreren Teilen zusammengesetztes Ganzes“) *Das Ganze ist mehr als die Summe seiner Teile: horizontale Unterteilung in Teilsysteme und vertikale Unterteilung in Untersysteme.* • Ein System wird bestimmt durch: Systemziele, Systemelemente Systembeziehungen, Systemverhalten

Organisation („ein soziales Gebilde, das dauerhaft ein Ziel verfolgt und eine formale Struktur aufweist, mit deren Hilfe Aktivitäten der Mitglieder auf das verfolgte Ziel ausgerichtet werden sollen“) *Schlüsselmerkmale: Ziel, dauerhaft Mitglieder, formale Struktur, Aktivitäten der Mitglieder* • Eine Organisation ist ein soziales System

Aufbauorganisation (Organisationsstruktur) •

Ablauforganisation (Organisationsprozess) •

instrumentale Organisation (Organisation als Werkzeug zur Zielerreichung) • Verteilung von Aufgaben und Kompetenzen, Abwicklung von Arbeitsprozessen.

institutionale Organisation (Organisation als Einrichtung („Abteilung“)) •

Standardisierung (Aufbau von Routine durch vorheriges Durchdenken und Festlegen von Verhaltensweisen) • Unterscheide: routiniertes Verhalten vs problemlösendes Verhalten.

Projekt (innovatives, komplexes Vorhaben mit definierten Rahmenbedingungen) *internes Projekt oder externes Projekt* • Eigenschaften, sprich Rahmen: Umfang, Dauer, Besonderheit Komplexität, Schwierigkeit, Bedeutung, Risiko, Kosten, Kontinuität, Intensität, Anzahl, Organisations- und Führungsverständnis

Projektaspekte () • betriebswirtschaftliche Sicht, technische Sicht, soziologische Sicht

Software Engineering (geprägt auf NATO-Konferenz 1968 in Garmisch/D) *ingenieurmäßige Softwareentwicklung* • Programmieren als Ingenieursleistung.

Software Projekt Engineering (ist die organisierte Softwareentwicklung in Form von Projekten.)

•

Projektentwicklungszyklus (K2)

Projektentwicklungszyklus (Typische Schritte)
• 1) Initialisierung, 2) Angebotslegung und Auftragserteilung, 3) Vorprojekt, 4) **Produkterstellung in Basisprodukt** gefolgt von Optimierung, Adaptierung, Erweiterung 5) Nutzung.

Initialisierung (erster Schritt Projektentwicklungszyklus) *Tipp: Auch internes Projekt wie externes behandeln!* • Zielformulierung schriftlich festhalten. Projektumfeld erkennen (Umweltanalyse). Mitarbeiter auswählen. Literatur, Expertenwissen zusammenstellen.

Angebot, Auftrag (zweiter Schritt) *Auftragserteilung immer schriftlich!* • unverbindliches Angebot oder verbindliches Angebot.

Vorprojekt (dritter Schritt) *Verfeinerung des Angebots zur besseren Schätzung der Machbarkeit, Kosten und Qualität.* • Istanalyse durchführen, Sollkonzept er-

stellen, Umfang der Machbarkeit festlegen, nicht im Ergebnis Enthaltene festlegen.

Basisprodukt (vierter Schritt (1)) *erfüllt die wichtigsten Grundanforderungen* • Muss-Anforderungen + Notwendigkeiten durch den Stand der Technik + technisch Notwendiges für Muss-Anforderungen aus Soll- und Kann-Anforderungen. Schrittweise vorgehen: Meilensteintechnik!

Optimierung, Adaptierung, Erweiterung (vierter Schritt (2)) • nach Abnahme des Basisprodukts, als Folgeprojekte.

Nutzung (fünfter Schritt) • Ein Projekt ist dann erfolgreich, wenn das erstellte Produkt vom Auftraggeber bzw. (dessen) Kunden genutzt wird!

Lebensdauer von Software (Software „lebt“ 18+ Jahre, Hardware 2-3!) • Zukunftssichere Techniken auswählen!

Produktentwicklungszyklus (K3)

Produktentwicklungszyklus (Aufgabentrennung) *Planung - Production - Checking* • Planung, Durchführung (setzt Planung voraus), Überprüfung (Planung ohne Überprüfung ist sinnlos, Überprüfung ohne Planung unmöglich!)

Planung (Planung bedeutet Steuerung einer Aufgabe. Erst durch die Überprüfung ergibt sich eine Regelung!) • Ziel: Vorgaben festlegen für: Aufgaben, Aufwand, Termine, Ressourcen,

Kosten, Finanzen, begleitende Maßnahmen (Qualitätssicherung, Risikobehandlung, ...)

Durchführung (schrittweise Produkterstellung)
• Ziel: schrittweise Produkterstellung. Reihenfolgeplanung: Hardest-first-Strategie, Easiest-first-Strategie oder Mischstrategie.

Überprüfung (Regelung des Produktentwicklungsprozesses) • Erkennung von Abweichungen und einleiten korrigierender Maßnahmen.

Ingenieurmäßige Projektentwicklung (K4)

Ingenieurmäßige Projektentwicklung (Wirtschaftliche Aufgaben) *Führung, Entwicklung, Kontrolle* • Keine Personalunion (bspw. Führung und Kontrolle)!

Führung (technische Führung, kaufmännische Führung (Periodenfenster!)) *organisieren, steuern, Personal führen (motivieren), informieren* • **Ziele:** Projektvision umsetzen, Projekt „auf Kurs halten“, technische, wirtschaftliche und soziale Rahmenbedingungen einhalten. **Aufgaben:** Ziele festlegen, Realisierbarkeit prüfen, Pläne erstellen, Pläne gemäß Überprüfungsergebnissen laufend adaptieren, bei mehreren gleichzeitig laufenden Projekten: Multi-Projektführung

Entwicklung (Einzelentwicklung, verteilte Entwicklung, serielle Entwicklung, parallele Entwicklung, konkurrierende Entwicklung) *Ziele setzen, planen, entscheiden, umsetzen* • **Ziele:** Projektziel gemäß Planung umsetzen, Planabweichungen erkennen und begründen. **Aufgaben:** Design erstellen und verfeinern, Prototypen (weiter-)entwickeln, Testsuite erstellen, Produktfeatures implementieren und testen,

Produktdokumentation erstellen

Kontrolle (Aufgabenkontrolle, Ablauf- und Terminkontrolle, Leistungskontrolle, Kosten- und Finanzierungskontrolle, Ressourcen- und Kapazitätskontrolle, Qualitätskontrolle) *überprüfen, überwachen* • **Ziel:** Planabweichungen aktiv erkennen (aktive Kontrolle notwendig!). **Aufgaben:** alle geplanten Bereiche kontrollieren, Art und Häufigkeit der Überprüfungen vorgeben. **Die integrierte Kontrolle berücksichtigt alle Faktoren und setzt die Abweichungen zueinander in Beziehung.**

Schwierigkeiten (Führung) • Einmaligkeit von Softwaresystemen, sehr technische Führungssicht, mangelhafte Planung, hohe Zahl von Lösungsmöglichkeiten

Schwierigkeiten (Entwicklung) • Individualität der Programmierer, rasche technologischen Veränderungen, Fehlen genormter Bauteile und Baugruppen

Schwierigkeiten (Kontrolle) • das Immaterielle von Softwareprodukten

Aufbauorganisation (K5)

Aufbauorganisation (beschreibt die Struktur eines Systems (engl.: „organization structure“)) • Unterscheide: allgemeine Organisationsformen, softwareentwicklungsspezifische Organisationsformen

Reine Projektorganisation (Allgemeine Organisationsformen) *ausschließlich projektbezogen („Taskforce“); Projektmitarbeiter werden aus der bestehenden Organisation herausgelöst* • **Vorteile:** eindeutige Aufgabenzuordnung, volle Konzentration auf Projekt, Identifikation mit der Aufgabe, rasche Entscheidungen, flexible Reaktion. **Nachteile:** Schwächung der Abteilungen, Abstellung der am leichtesten entbehrlichen Mitarbeiter („Mr. Bean-Effekt“), Konflikte mit angestammten Abteilungen, unterschiedliche Auslastungen. **Geeignet für schwierige, kleine Projekte.**

Einfluss-Projektorganisation (allg. O.F.) *funktionale Hierarchie unverändert; Projektleiter als Stabsfunktion* • **Vorteile:** Abteilungsorganisation unverändert, Sicherheitsgefühl für Mitarbeiter. **Nachteile:** mangelnde Autorität des Projektleiters, Arbeiten am Projekt nur, „wenn Zeit ist“, Konflikte werden verzögert. **Geeignet für gut strukturierte, inhaltlich einfache Projekte.**

Matrix-Projektorganisation (allg. O.F.) *zweidimensionale Organisationsform (Unternehmenshierarchie und Projekt) mit doppelten Vorgesetzten („Matrix“). Beabsichtigter Konflikt.* • **Vorteile:** selbstständiger Projektleiter, Mitarbeiter bleiben in Abteilung Expertenwissen leicht einbaubar. **Nachteile:** Verunsicherung bei Vorgesetzten, Verunsicherung bei Mitarbeitern, man

„sitzt zwischen zwei Stühlen“, Kompetenzkonflikte, Möglichkeit: Vorgesetzte auszuspielen. **Geeignet für zeitunkritische Projekte.**

Hierarchische Teamorganisation (sw-spez. O.F.) *Zerlegung in Phasen und Einrichtung entsprechender Abteilungen und Leiter (z.B. Analyse, Programmierung, Testen).* • **Vorteile:** entspricht anderen Industriezweigen. **Nachteile:** Projektleiter weit vom Geschehen, entfernt langsame indirekte Kommunikation, Peterprinzip! **Für Softwareentwicklung zu bürokratisch und undemokratisch**

Chefprogrammierer-Team (sw-spez. O.F.) *dem Chirurgen-Team nachgebildet [Baker, 1972]; kleine Teamgröße, Spezialisten, Projektleiter arbeitet aktiv mit.* • **Vorteile:** Chefprogrammierer direkt eingebunden, gute Kommunikation, Berichtswesen institutionalisiert. **Nachteile:** nur kleine Teams möglich, Personalanforderungen kaum erfüllbar, undankbare Aufgabe des Projektsekretärs. **Es wurde gezeigt, dass kleine Teams besser sein können als Pgm.-heere!**

Moderne Projektgruppe (sw-spez. O.F.) *Änderungen in der Softwareentwicklung erfordern Anpassungen des Chefprogrammierer-Team-Konzepts* • Softwareingenieur muss alle Schritte des Entwicklungsprozesses beherrschen, Mitarbeiter müssen Kollegen vertreten können. Einbindung in die Organisation: als projektorientierte Teilorganisation, durch Projektleitung in der Linienorganisation, bei mehreren kooperierenden Gruppen: Leitergruppe.

Gruppendynamische Prozesse (Gruppendynamische Prozesse führen zur Selbstordnung) •

Kommunikation der Mitglieder sowie deren Koordination durch den Projektleiter ist der entscheidende Erfolgsfaktor. Gruppendynamische Prozesse führen zur Selbstordnung. Hohe Ethik und Moral notwendig.

Motivation der Mitarbeiter (Software Engineering ist eine humanzentrierte Tätigkeit!) • realistische Planvorgaben, klare Kompetenzen, Gefühl individueller Wichtigkeit, akzeptable Arbeitsbedingungen, auch privat gutes Gesprächsklima notwendig, Konflikte ansprechen und beheben! (Gewinner und Verlierer) .

Parkinsonsches Gesetz (Man braucht immer (mindestens) so viel Zeit, wie man hat!) • Dauernde

leichte Überforderung führt zu besserer Arbeit!

Produktivität (Nutzleistung und Blindleistung) *Ein Projekt, das in Verzug ist, gerät durch Hinzufügen neuer Mitarbeiter nur noch mehr in Verzug!* • Produktivität einer Projektgruppe lässt sich unterteilen in produktive Arbeit (Nutzleistung) und Kommunikation (Bindleistung). Optimale Gruppengröße siehe: Kapitel 5 # 23 und **Entwicklungszeit in der Gruppe (Brooksches Gesetz)** Kapitel 5 #24

Organisationsmängel (Hauptmängel) • Fehlen von Planung, Organisation, Standards Inkompetenz der Projektführung, Fehlen aktueller Dokumentation, Mangel an Fortschritts- und Qualitätskontrolle, Mangel an Kostenkontrolle

Ablauforganisation (K6)

Ablauforganisation (Aneinanderreihung systeminterner Elemente (Arbeitsabläufe) zur Zielerreichung) • Arbeitsvorgänge organisieren, Verantwortlichkeiten formalisieren, Rationalisierung ermöglichen, Standardisierung anstreben. **Aufgaben:** Prozesse in Gang setzen, Prozesse analysieren, Prozesse verbessern. Prozesse sind produktbezogen (aus Projektsicht statisch) oder prozessbezogen (aus Projektsicht dynamisch).

Prozessdefinitionen ([IEEE]) • Sequenz von Schritten zur Zweckerfüllung.

Gesteuerter Prozess (keine Rückkopplung) • Eingabe, Verarbeitung, Ausgabe

Geregelter Prozess (Rückkopplung) • Eingabe, Verarbeitung, Soll/IstVergleich (Rückkopplung, Eingabe), Ausgabe. **In der Softwareentwicklung benötigt man geregelte Prozesse!**

Definierter Prozess (z.B. Schrittweise Verfeinerung) • Aktivitäten vorab bekannt und verstanden, Ergebnis vorhersehbar und meist reversibel, Prozess wiederholbar, Beginn und Ende vorab festlegbar

Empirischer Prozess (z.B. iterativ-inkrementelles Adaptieren) • komplex und unvorhersehbar, nicht vollständig verstanden, nicht durchgehend definierbar, „erwarte das Unerwartete!“. **Auch empirische Prozesse können erfolgreich geregelt werden, selbst wenn sie nicht vollständig verstanden werden („kreativer Teilprozess“)!** Erfolgreiche Regelung durch: häufige Inspektionen, laufende Anpassung, Management der Störfaktoren. **Prozesse in der Softwareentwicklung werden im 21. Jh. als empirische Prozesse gesehen!**

Value-added Software Development (Rückkopplung durch Bewertung der Zwischenprodukte durch den Kunden ist entscheidend für den weiteren Projektverlauf (emergentes Verhalten) sowie die Ausrichtung des Produkts (Value-added Software Development).) • Paradigmen siehe K6S11

Projekt-Lebenszyklus (Ordnung der zeitlichen Abfolge der Aktivitäten) • von der Entwicklung bis zum Einsatz

Software-Lebenszyklus (Ordnung der zeitlichen Abfolge der Aktivitäten) *von der Entwicklung über den Einsatz bis zum Ende der Benutzung*

•

Anwendungs-Lebenszyklus-Management (ALM - Trennung vor/nach Projektende wird unscharf) • kontinuierliche Entwicklung und Freigabe während des gesamten Lebenszyklus des Produkts.

Vorgehensmodell (traditionell oder agil) • ALT: Dokumentation der Ablauforganisation in der SW-Entwicklung wurde Ende der Neunziger Jahre immer umfangreicher. NEU: Systematische Gliederung einer Lösung. Projektmanager versuchen zunehmend, aus Prozess Erfahrungen zu lernen. Prozessmuster (Patterns) sind bewährte Praktiken, die induktiv aus Prozess Erfahrungen erarbeitet wurden. Es gibt auch Sammlungen von Negativbeispielen (Anti-Patterns).

Vorgehensmethoden (Vorgehensprozess durch Vorgehensmethoden) • Unterscheide: Vorgehensprozess (engineering process); Umsetzung durch Vorgehensmethoden (engineering practices). Oft Erfolg nur durch Kombination verschiedener Methoden!

Sequenzielles Vorgehen (Vorgehensmethoden) • Phasenmodell. begleitet von der Organisation der Entwicklung allgemeiner technischer Systeme, veraltet.

Inkrementelles Vorgehen (Vorgehensmethoden) K6S23 • **Wasserfallmodell:** kaum Flexibilität, Änderungswünsche teuer; **Objektorientiertes Modell:** Tool-Smithing“, **Wiederverwendung** setzt **Wiederverwendbarkeit** voraus!

Iteratives Vorgehen (Vorgehensmethoden) K6S26 • Grundidee: kontinuierliche Verbesserung (PDCA-Zyklus) [Deming] (iterativ = wiederholt!). Erstellung eines neuen Produkts in jeder Iteration, aber Aufbauen auf erarbeitetem Wissen! Motivation: Fehlerbehebungskosten steigen im Laufe der Projektentwicklung überproportional an. Siehe **Exploratives Prototyping** K6S29, **Spiral-Model** K6S30

Iterativ-inkrementelles Vorgehen (Vorge-

hensmethoden) K6S31 • Evolutionäres Vorgehen – Entwicklung in Zyklen, jeweils auf den Ergebnissen des vorherigen Zyklus aufbauend. Beispiel: Scrum. **Evolutionäres Prototyping:** Pro-

totypen werden inkrementell weiterentwickelt (K6S32). **Schablonenmodell:** Erkennen und Anwenden von Denkmustern (patterns) und Problemlöseschablonen (templates).

Informationsaustausch (K7)

Kommunikation (K7S2)

Kommunikation in Softwareprojekten (mündlich wie auch schriftlich – alles Wichtige schriftlich festhalten!) *Trends: Kundeneinbindung steigt (agiles Vorgehen), räumliche Distanz der Mitarbeiter steigt, Komplexität von Werkzeugen und Ergebnissen steigt, Entwicklungszeit sinkt* •

Stakeholder (alle Beteiligten und Betroffenen eines Projekts) • Auftraggeber / Kunde(n), Auftragnehmer (Eigentümer), Anwender, Entwickler, Projektleiter, Sub-Auftragnehmer, Zulieferer, Investoren, Fördergeber

Value-Based Software Engineering (VBSE) (Bewertung der Anforderungen führt zu höherer Kundeneinbindung.) • Dies bedingt: kompetenten und entscheidungsbefugten Auftraggeber gute Werkzeugunterstützung der Anforderungsanalyse (traceability).

Besprechungen (Interne Projektbesprechungen (Fortschritt) oder Externe Projektbesprechungen (AG)) • Besprechungen sind zu protokollieren (Stichwortprotokoll). Für jede Besprechung ist die Beginn- und Endzeit vorher bekannt zu geben. **Besprechungsleiter:** K7S8

Reviews (formale Überprüfungen von Zwischenergebnissen oder Ergebnissen) *Bei Reviews ist der Erfolg stark von der Akzeptanz bei den Gruppenmitgliedern*

abhängig! • in Gruppen durchgeführt, breit einsetzbar (Dokumente, Benutzerschnittstelle, Quellcode, ...), auch für halb fertige Produkte anwendbar, Informationsaustausch in der Gruppe wird gefördert, verschiedene Teamzusammensetzungen fürs Review möglich.

Computer-Aided-Software-Engineering-Systeme (CASE-Systeme) (Werkzeuge (K7S11)) • sind breit einsetzbare Werkzeuge oder eine koordinierte Sammlung von Werkzeugen zur Entwicklung von Softwareprodukten oder -services..

Application-Lifecycle-Management-Systeme (ALM-Systeme) (Werkzeuge (K7S11)) • sind Werkzeuge, die neben der Entwicklung von Software auch den Betrieb (Wartung, Pflege) dieser Software unterstützen. ALM-Systeme lösen zunehmend CASE-Systeme ab.

Softwareentwicklungsumgebungen (IDE) (Werkzeuge (K7S12)) *Werkzeugnutzung ist gerade bei agilem Vorgehen essenziell!* • sind integrierte Entwicklungssysteme. Sie umfassen: Planung („Projektmanagement“), Prototyping Analyse, Design („Roundtrip-Engineering“), Implementierung, Testfallgenerierung, -durchführung, Konfigurations- und Versionsverwaltung, Dokumentation.

Dokumentation (K7S15)

Dokumentation von Softwareprojekten (umfasst Projektdokumentation („organisatorische Dokumentation“) und Produktdokumentation („technische Dokumentation“)) • Ziele: einheitliche Dokumentstruktur, rasche Erstellung neuer Dokumente (Vorlagen), einfache Zusammenführung von Dokumenten, aktuelle Information über Projektstand (Projektkontrolle). Aufgabe: Erstellung von Standards, Strukturen und Vorgaben

Berichtswesenplanung (K7S20) *Wann ist es zu erstellen? Wer erstellt es? Wer empfängt es? Wie oft ist ein Update notwendig? Sollen Versionen gehalten werden?*

Welches Format ist zu wählen? • Die Berichtswesenplanung legt den Kontrollfluss für Informationen fest, d. h. wer von wem wann welche Informationen erhält)

Dokumentationsorganisation (K7S22) *Ablageort? (lokal, Intranet, Cloud,...) Ablageformat? (Originalformat, strukturiert – XML, PDF/A,...) Indexierung für Suche (Beschlagwortung, Volltextzugriff, ...) Zugriffsschutz? Sekundärsicherung? (z.B. extern) Wartung der Ablage? (z.B. Transferieren auf neues Medium alle 5 Jahre) Ablagedauer? (Entsorgung bzw. sicheres Löschen danach)* • für sicheres Ablegen und rasches Wiederfinden von Dokumenten

Konfigurationsmanagement (K7S24)

Konfigurationsmanagement (Jedes Softwareprojekt braucht eine Konfigurationsverwaltung ... händisch oder werkzeugunterstützt! (K7S26+)) **Versionsverwaltung:** Versionen von

Dokumenten und Code, Zugriffsrechte (Teamprojekte!), **Konfigurationsverwaltung:** Revisionen, Varianten, Releases, **Build-Unterstützung:** Continuous Integration, Continuous Release, **Workflow-**

Unterstützung, Änderungsmanagement: Anforderungen, Fehler, Erweiterungen, **Auswertungen** • „...ist derjenige Aufgabenbereich in der Projektentwicklung, der für die Verwaltung und Speicherung aller innerhalb des Projekts anfallenden Zwischen- und

Endergebnisse und deren Versionen verantwortlich ist.“ [IEEE]

Konfigurationsmanagement-Verantwortlicher (K7S2) • Betreuen des GIT/TFS/etc

Dokumente der Aufbauorganisation (K7S29)

Organigramm (K7S40) • Darstellung von Gesamtstruktur und Stellen (Funktionen) in der Aufbauorg

Stellenbeschreibung (K7S41) Eine Stellenbeschreibung enthält: Stellenbezeichnung, Aufgaben der Stelle, notwendige Qualifikationen, Verweis auf Vorgesetzte(n),

Regelung der Stellvertretung • Beschreibung von Aufgaben, Kompetenzen und Verantwortlichkeiten zur Schaffung klarer Zuständigkeiten. Eine Stellenbeschreibung wird je **Stelle erstellt** (NICHT je Mitarbeiter/Rolle)!

Dokumente der Ablauforganisation (K7S43)

Grundlegende Dokumente (K7S44) • Ziel: einheitliche Gliederung der Dokumente

Projekthandbuch (K7S45) Organigramm, Stellenbeschreibungen, Stellenzuordnungen, Projektleitung, Führungsgrundsätze, Sitzungskonzept, Dokumentationsvorgaben, -standards, Vorgehensmethode(n), Werkzeuge, Vorgaben für Projektplanung (Arten, Umfang, Häufigkeiten), Festlegungen zur Qualitätssicherung • verbindliche Festlegung der Projektorganisation.

Projekttagbuch (K7S46) Ein Eintrag enthält: Datum, (Uhrzeit). Beteiligte (Ressourcen), Aktivität,

Aufwand • entwicklerinterne, chronologische Aufzeichnung aller Projektstätigkeiten. Das Projekttagbuch wird von allen Projektmitarbeitern gemeinsam geführt und ist laufend (täglich) zu aktualisieren.

Projektfortschrittsbericht (K7S47) • Terminbericht zum Zweck des Soll-Ist-Vergleichs, ergeht von Projektleiter AN an Projektleiter AG oder Projektträger, ist regelmäßig zu erstellen (z.B. alle 4 Wochen) sowie im Anlassfall (z.B. Revision)

Arbeitsbericht & Projektbibliothek (K7S47+) • siehe Folien.

Qualitätssicherung (K8)

Ziel

Qualität (Jeder Prozessschritt ist für die Qualität verantwortlich!) *Qualität (im Unterschied zur Spitzenqualität) bedeutet also die Erfüllung der Anforderungen (nicht mehr, nicht weniger!).* • Definition gemäß ISO 9000:2005: „[Quality is the] degree to which a set of inherent characteristics fulfils requirements“. („[Requirements are] needs or expectations that are stated, generally implied or obligatory“).

Qualitätssicherung (K8S13) • Sicherung der Produktqualität über den ganzen Projektentwicklungsprozess; „... das Erzeugen von Vertrauen darauf gerichtet... , dass Qualitätsanforderungen er-

füllt werden“ [ISO 9000:2005]

Priorisierung der Qualitätsmerkmale (K8S11) *Ziel: Qualitätsgesteuerter Softwareentwicklungsprozess* • Maßnahmen zur Erfüllung von Qualitätsanforderungen erhöhen den Erfüllungsgrad bestimmter Qualitätsmerkmale, können aber andere Merkmale negativ beeinflussen. Merkmale priorisieren!

Festlegungen Qualitätsmerkmal (K8S12) • Definition (im Projektkontext), Priorität, Maß, Maßeinheit, Messvorgang, Schwellenwert.

Aufgaben (K8S15)

Qualitätsplanung (K8S16) • Festlegen der Qualitätsmerkmale, laufendes Adaptieren **Mehr K8S18.**

Qualitätsprüfung (K8S16) • Prüfen der Projektgrundlagen (Bausteine, zugekaufte Komponenten),

laufendes Prüfen der Zwischenergebnisse. **Mehr K8S19**

Qualitätslenkung (K8S16) • Anwenden von Maßnahmen zur Qualitätssicherung **Mehr K8S20**

Techniken (K8S22)

Techniken der Qualitätssicherung (K8S22) *Einhalten einer Norm oder Verwendung eines Modells/Verfahrens zur Qualitätssicherung heißt nicht automatisch hohe Qualität, sondern nur Offenlegung der Qualität des (Produktions-)Prozesses.* • Techniken der Qualitätssicherung beziehen sich nicht auf das Pro-

dukt, sondern (vorrangig) auf den Prozess. Beispiele für Modelle: CMMI, EFQM Beispiele für Normen: ISO 9000 ff., ISO/IEC 15504 „SPICE“, ISO 25000 ff. Beispiele für Verfahren: Balanced Scorecard, Six Sigma, Kontinuierlicher Verbesserungsprozess (KVP)

Ergebnisse (K8S25)

Qualitätshandbuch (K8S25) *in Europa vielfach nach ISO 9000 ff. strukturiert* • schriftliche Dokumentation des Anwendungsbereichs der Qualitätssicherung (Ausschlüsse begründen), der Ver-

fahren zur Qualitätssicherung (Planung, Prüfung und Lenkung), des Zusammenspiels der Qualitätssicherungsprozesse

Normen und Richtlinien (K8S26)

Norm (K8S27) • durch (über-)staatliche bzw. davon beauftragte Einrichtung erstellt, in bestimmten Fällen Gesetzescharakter (verpflichtend!), Verbindlichkeit oft auch ohne explizite Erwähnung im Projektvertrag

Richtlinie (K8S27) • auch durch Unternehmen(sverbände), Interessensgruppen etc. erstellt, darf verpflichtenden Normen nicht widersprechen, Verbindlichkeit nur bei expliziter Erwähnung im Projektvertrag

Zielerforschung (K9)

Zielerforschung (Festlegung des Gesamtziels (der Grobziele)) *Dieses darf sich während des Projekts nicht ändern! (Anforderungen ändern sich)* • Bei Änderung von Grobzielen Projektabbruch, (Herausforderung bei schleichender Änderung)!

Die Erhebung

Erhebung - Ziel (K9S6) • Feststellung der Projektausgangslage durch den Auftragnehmer im Umfeld des Auftraggebers

Erhebung - Aufgaben (K9S7) • Grobziele des Projekts (schriftlich) formulieren, Projektvorarbeiten zusammenstellen, Projektumfeld definieren, Stakeholder ermitteln

Erhebungstechniken (K9S10)

Art der Interaktion (Erhebungstechniken) • Einteilung nach der Art der Interaktion: auditiv (Interview) oder schriftlich (Fragebogen)

Strukturierungsgrad (Erhebungstechniken) • Einteilung nach dem Strukturierungsgrad: nicht strukturiert (freies Gespräch), halb strukturiert (Fragen vorgegeben), strukturiert (Fragen und Antworten vorgegeben)

Objekt der Erhebung (Erhebungstechniken) • Einteilung nach dem Objekt der Erhebung: Primärerhebung oder Sekundärerhebung (Dokumentenauswertung)

Anzahl der Erhebungsobjekte (Erhebungstechniken) • Einteilung der Anzahl der Erhebungsobjekte: Einzelerhebung Fallstudie repräsentative Erhebung Totalerhebung

Erh.-Techniken - Interview (K9S12)

Interview (Interaktionsprozess zwischen einzelnen Personen oder Gruppen) *verschiedene Fragetypen geschickt nutzen: aufschließende / abschließende Fragen offene / geschlossene Fragen direkte / indirekte Fragen* • **Nicht strukturiertes Interview:** offenes Gespräch zwischen Interviewer und Interviewtem (trotzdem Interviewleitfaden erstellen), **Halb strukturiertes Interview:** Fragen vorgegeben, **Strukturiertes Interview:** Fragen und Antwortmöglichkeiten vorgegeben

Erh.-Techniken - Fragebogen (K9S13)

Fragebogen (Schrittweises Vorgehen) • 1. Zielgruppe auswählen 2. Fragen formulieren (KROKUS*-Regel) 3. Fragebogen zusammenstellen 4. Fragebogen testen 5. Fragebogen einsetzen 6. Fragebogen auswerten

KROKUS*-Regel (Fragen formulieren) • Kurze Fragen stellen – Redundante Fragen vermeiden – Offene Fragen stellen – Konkrete Fragen stellen – Unter- und Kettenfragen vermeiden – Suggestive Fragen vermeiden

Erh.-Techniken - Dokumentenauswertung (K9S15)

Dokumentenauswertung (Auswertung von Sekundärquellen, da (auftraggeberinterne) Projektvorarbeiten oft schwierig zugänglich) *Beispiel: Produktvergleich* • Wichtig: Entscheidungen bewusst und nachvollziehbar machen, Gewichtung für den

Auftraggeber transparent machen und änderbar gestalten, Plausibilitäts- und Sensibilitätsüberprüfungen durchführen

Ergebnisse der Erhebung (K9S18)

Erhebungsauswertung (Erhebung – Ergebnisse) • Bericht des Auftragnehmers an den Auftraggeber beschreibt Problem aus der Sicht des Auftragnehmers, klärt Verständnis der Grobziele ab, ist nicht bindend.

Die Zielfindung (K9S21)

Zielfindung (Ziel) • Erarbeitung des Gesamtziels und der verbundenen Grobziele auf Basis der Erhebung

Zielfindung (Aufgaben) *Für jedes Ziel muss der Erreichungsgrad bestimmbar und überprüfbar sein* • Ziele gewichten und klassifizieren (Muss, Soll, Wunsch), Beweglichkeiten der Ziele analysieren, Organisationsvorgaben festlegen, unterschiedliche Zielarten (technische Ziele, organisatorische Ziele, finanzielle Ziele, ...) herausarbeiten, erste Lösungsskizze ermitteln

Zielfindungstechniken (K9S25)

Kreativitätstechniken („schöpferisch“) *Einsatz: zur iterativen und interaktiven Lösungsfindung, zur Findung von Lösungsalternativen* organisiert als Sitzungen heterogener Gruppen („interdisziplinäre Kreativitätsgruppen“); teilweise auch alleine möglich, gesteuert und dokumentiert durch ModeratorInnen • Methoden zur Definition und Lösung schlecht strukturierter Probleme mittels intuitiver Probierv Verfahren.

Vorgehen bei Kreativitätstechniken ((Kreativitätstechniken)) • 1 Problem in Teilprobleme zerlegen 2 Lösungsvorschläge generieren 3 beste Teillösungen auswählen 4 Problemlösung zusammensetzen

Technik - Brainstorming (K9S27)

Voraussetzungen (Brainstorming) • präzise Problemdefinition, geringe Komplexität

Vorgangsweise (Brainstorming) • 1. Ideengenerierungsphase (15 min; möglichst viele originelle Ideen, keine Kritik), 2. Ideenbewertungsphase (10 min; Selektion brauchbarer Ideen)

Vorteile (Brainstorming) • viele Ideen in kurzer Zeit, sehr unterschiedliche Ideen, einfaches Verfahren

Nachteile (Brainstorming) • mangelnde Autorität des Projektleiters, realitätsfremde Ideen, Abdriften vom eigentlichen Problem

Technik - Brainstorming /Osborne-V. (K9S29)

Vorgangsweise (Osborne Verfremdung) *Ideenbewertungsphase bleibt unverändert* • Erweiterung der Ideengenerierungsphase: systematische Abwandlung der Problembeschreibung, Erhöhung möglicher Lösungen

Technik - Synektik (K9S30)

Idee (Synektik) • spontane Problemdefinition mit Nachfragen, Wer?, Warum?, Wie?, Wozu?, ... - „W-Technik“

Vorgangsweise (Synektik) • 1. Spontandefinition 2. Umformulierung (Warum-Frage) 3. Antwortenermittlung 4. Problemdefinition 5. Eigenschaftsprüfung 6. Rückkopplung, **Abbruch bei Konvergenz**

Vorteile (Synektik) • zielgerichtetes Vorgehen, auch für größere Probleme anwendbar

Nachteil (Synektik) • mehrere Fachleute notwendig

Technik - Morphologischer Kasten (K9S32)

Idee (Morphologischer Kasten) • Parametrisieren des Problems mit Analyse des Lösungsraums

Vorgangsweise (Morphologischer Kasten) • 1. präzise Problemdefinition, 2. Zerlegung in Problemelemente, 3. Lösungssuche pro Element, 4. Kombination und Bewertung der Lösungen, 5. Auswahl und Realisierung der besten Lösung

Vorteile (Morphologischer Kasten) • systematisches Vorgehen, keine voreiligen Entscheidungen, klare Lösung

Nachteil (Morphologischer Kasten) • Unübersichtlichkeit Auswahl optimaler Lösung schwierig

Ergebnisse (K9S38)

Zielbeschreibung (Zielfindungs-Ergebnis) • Bericht des Auftragnehmers an den Auftraggeber, beschreibt Gesamtziel als Problemdefinition mit skizzierter Lösung (keine lösungsbezogene Strukturierung), stellt Auftragnehmersicht dar > kein ursächlich bindendes Dokument

Datenlexikon (Zielfindungs-Ergebnis) *FDatenverzeichnis, Datenkatalog, Data Dictionary, Repository* • Verzeichnis aller projektrelevanten Fachbegriffe (alphabetisch sortiert), für jeden Begriff Definition / Erklärung, oft von Entwicklungsumgebungen bzw. ALM-Systemen unterstützt, **Wichtig: Datenlexikon laufend aktualisieren**

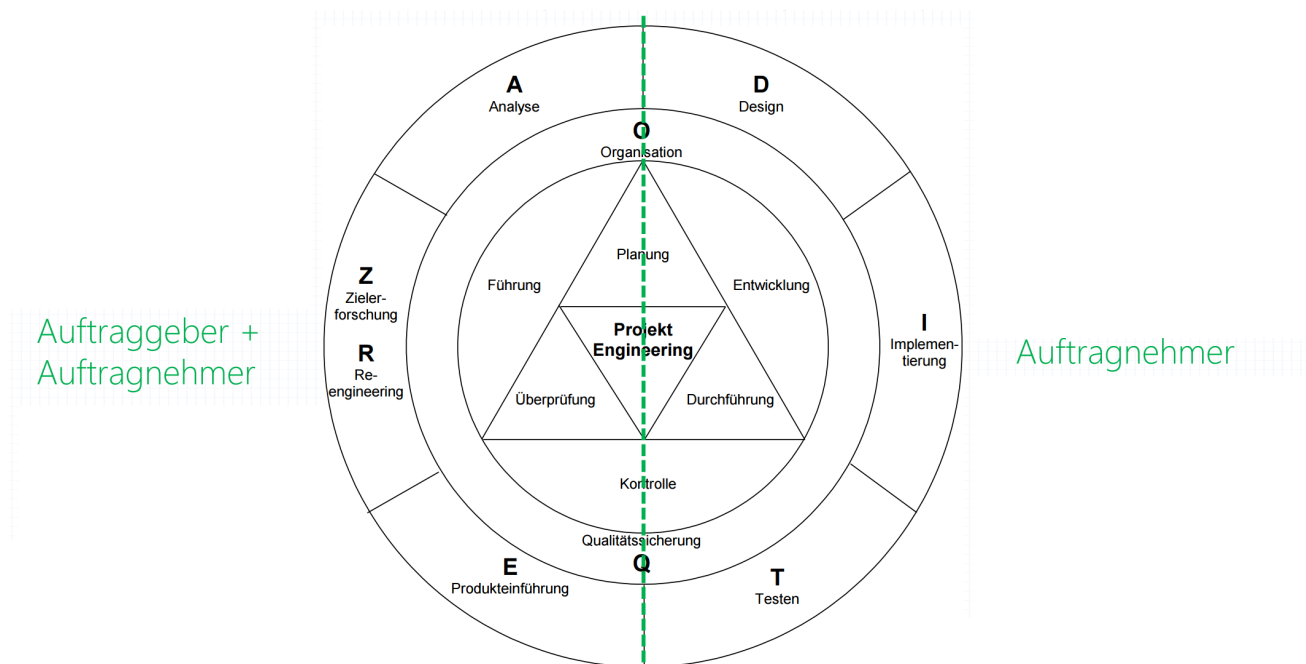


Figure 1: Projektentwicklung (K4)

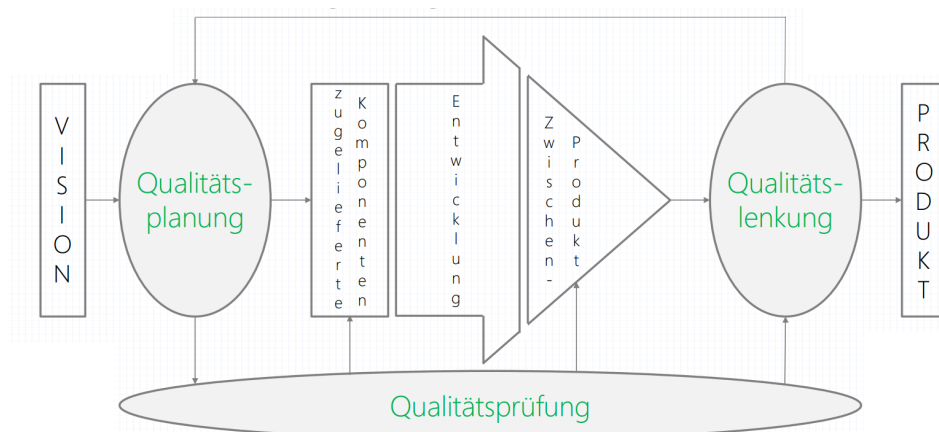


Figure 2: Qualitätssicherung (K8)

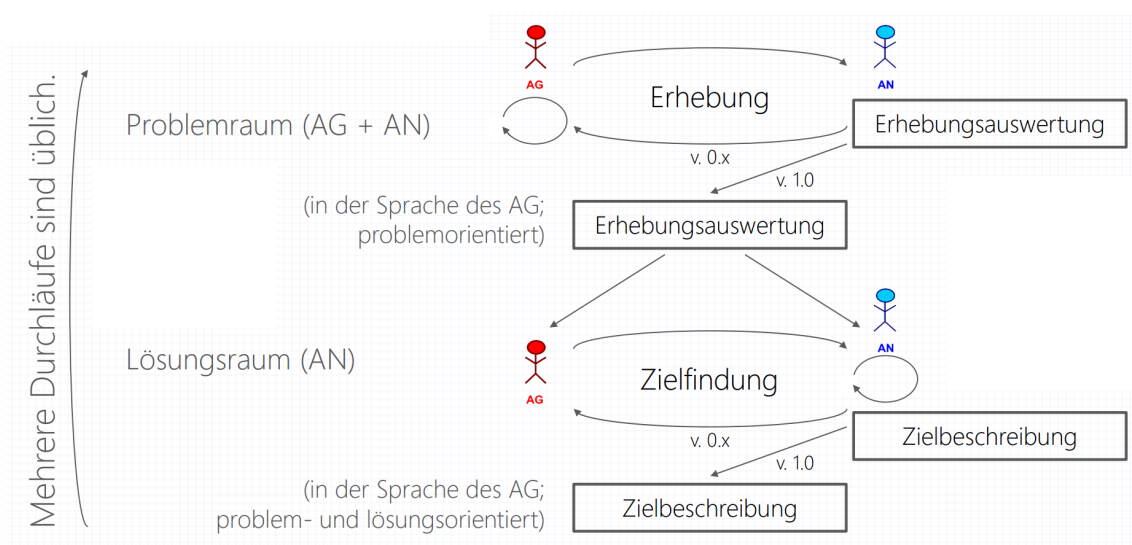


Figure 3: Zielforschung (Ablauf) (K9)