



Software- Engineering für langlebige Systeme

Projektmanagement in Projekten für langlebige Software

VL5

- Entscheidungen treffen und dokumentieren
- Managementfehler, die Softwareerosion fördern
- Wie man Managementfehler vermeidet
- Ziele:
 - Selber vollständige Entscheidungen treffen und dokumentieren können.
 - Managementfehler erkennen und vermeiden/beheben.
 - Strukturiertes Arbeiten im Projekt
- Allgemeine Grundlagen der Projektorganisation, -steuerung und -durchführung werden voraussetzt.
- Es werden besonders wichtige Punkt in Bezug zu langlebiger Software und Softwareerosion behandelt.

Entscheidungsfindung

Entscheidungsfindung und Dokumentation

- In diesem Block geht es nicht um alltägliche Entscheidungen, sondern um grundlegende Entscheidungen, die sich nur unter großem Aufwand revidieren lassen.
- Diese Art von Entscheidungen haben bei langlebigen System häufig größere Auswirkungen als bei „normalen“ Systemen.
- Problem:
 - Nachher weiß man es besser
 - Entscheidungen nicht nachvollziehbar
 - Entscheidungen häufig aufgrund von unvollständiger Analyse

Vorgehen

1. Aufstellen der Kriterien
2. Sammeln der Optionen
3. Aufstellen einer Bewertungstabelle
4. Auswerten der Tabelle
5. Qualitätskontrolle und evtl. Verfeinerung der vorherigen Schritte

Aufstellen der Kriterien

- Kriterien
 - Sammeln
 - Beschreiben
 - Wertungssystem festlegen
 - Durch Werte
 - Häufig: 1, .75, .5, 0
 - Verwenden von messbaren Werten
 - Kürzel
 - Festlegen von
 - K.O.-kriterien
 - Gewichten

Kriterium “Geschwindigkeit des Algorithmus” (GdA)

Da die Ergebnisse der Analyse ständig berechnet werden sollen, ist eine kurze Laufzeit des Algorithmus wichtig. Die Laufzeitanalyse soll aufgrund von Testläufen unter standardisierten Bedingungen auf gleichen Daten durchgeführt. . . . Bei einer Datenmenge vom 5000 Objekten wird das Ergebnis wie folgt eingestuft:

Wert	Wert num.	Laufzeit
sehr gut (++)	1	< 0.5s
gut (++)	0.8	< 1s
ausreichend (o)	0.5	< 2s
schlecht (-)	0	> 2s

Sammeln der Optionen

- Die Menge der Optionen soll vollständig sein
- Immer Default-Optionen einbeziehen (themenbezogen)
 - Nichts-Tun
 - Projektabbruch
 - ...
- Default-Optionen werden häufig ignoriert, obwohl sie regelmäßig die beste Wahl wären!

Aufstellen einer Bewertungstabelle

- Kriterien zu Optionen in Bezug setzen
 - Messen der Kriterien
 - Zu jedem Feld (Wertung) soll die Messung dokumentiert sein

Kriterium	...	GdA	Term	...	Punkte
Gewicht	...	2	K	...	Max: 11
Algorithmus I	...	+	+	...	9
Algorithmus I	...	–	–	...	K
⋮	⋮	⋮	⋮	⋮	⋮
Algorithmus n	...	–	+	...	7

Auswerten der Tabelle

- Zusammenrechnen der Werte
- Bestimmen der Gesamtpunkte
- Erfüllungsgrad bestimmen

Kriterium	...	GdA	Term	...	Punkte
Gewicht	...	2	K	...	Max: 11
Algorithmus I	...	+	+	...	9
Algorithmus I	...	-	-	...	K
⋮	⋮	⋮	⋮	⋮	⋮
Algorithmus n	...	-	+	...	7

Qualität des Ergebnisses

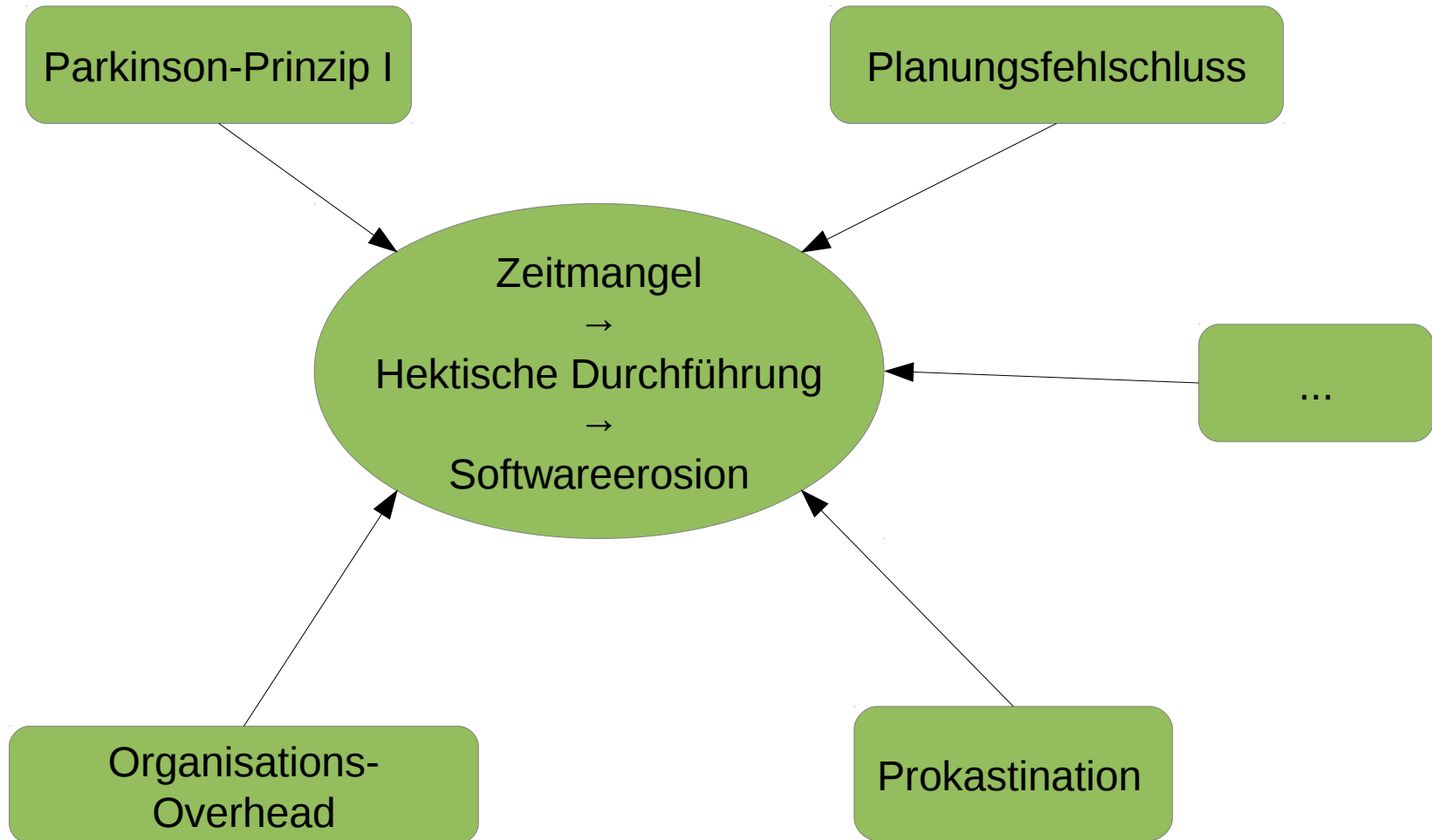
- Je geringer die prozentuale Übereinstimmung, je schlechter ist die Entscheidung
 - Nach weiteren Möglichkeiten suchen
 - Nicht an den Kriterien „drehen“!
 - Richtlinie: >80%

- Der Gewinner soll mir einem Abstand gewinnen
 - Gewichtungen und Wertungen sind immer ungenau
 - Bei geringen Abstand (z.b. 5%) kann eine unterlegende Möglichkeit der eigentliche Gewinner sein
 - Genauer Untersuchung der Top-Optionen
 - Mehr Kriterien definieren

Planungsfehler



Grundproblem



Planungsfehlschluss

Die benötigte Zeit für eine Aufgabe wird unterschätzt wobei der Wert der Aufgabe überschätzt wird

- Quelle: Scheinwerferprinzip, Induktion

Parkinson-Prinzip I

- Arbeit dehnt sich in genau dem Maß aus, wie Zeit für ihre Erledigung zur Verfügung steht, und nicht in dem Maß, wie komplex sie tatsächlich ist.

Parkinson-Prinzip II

- In Besprechungen werden die Themen am ausführlichsten diskutiert, von denen die meisten Teilnehmer Ahnung haben und nicht die Themen, die am wichtigsten sind.

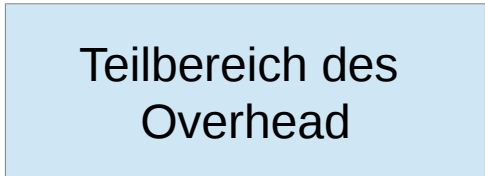
Parkinson Leitsätze

- Jeder Vorgesetzte wünscht die Zahl seiner Untergebenen, nicht jedoch die Zahl seiner Rivalen zu vergrößern
 - Machtwunsch
 - Sicherung der eigenen Position

- Mitarbeiter machen sich gegenseitig Arbeit
 - Verwaltung und Abläufe werden kompliziertet
 - Bei Problemen werden Steuerungsmethoden hinzugefügt, aber nie welche entfernt

Fehler bei Projektberechnungen

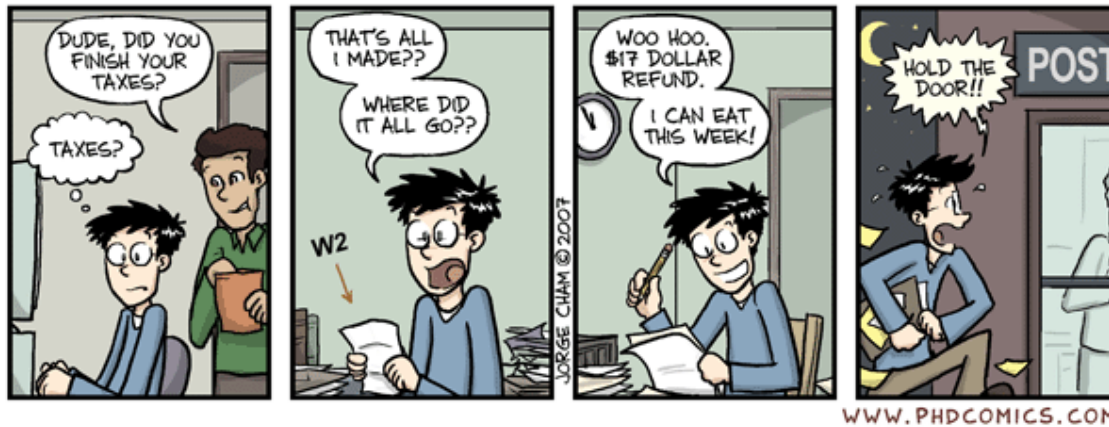
- Personen-Jahr/Monat/Woche/Tag/Stunde werden häufig direkt als Zeitansatz verwendet.
- Häufig werden bei der Umrechnung wichtige Faktoren vergessen:
 - Urlaub
 - Krankheit
 - Kommunikationsaufwand in Gruppen
 - Administrative Arbeiten
- Häufig wird nur 70-80% der wöchentlichen Arbeitszeit an der gemessenen Aufgabe gearbeitet.
- Werte schwanken von Land zu Land (2008):
 - D: ein Personenjahr = 1432 Stunden
 - USA: ein Personenjahr = 2087 Stunden
 - GR: ein Personenjahr = 2120 Stunden



Teilbereich des
Overhead

Procrastination

NOTHING IS CERTAIN EXCEPT PROCRASTINATION AND TAXES



"Piled Higher and Deeper" by Jorge Cham www.phdcomics.com

Procrastination

- wissenschaftliche Bezeichnung für pathologisches Aufschiebeverhalten
- Probleme in der Prioritätensetzung,
- mangelnde oder unrealistische Planung,
- Schwierigkeiten in der Abgrenzung gegen alternative Handlungstendenzen,
- Defizite im Zeitmanagement oder in der Konzentrationsfähigkeit,
- Abneigung gegen die Aufgabe,
- Angst vor Versagen oder Kritik,
- Fehleinschätzungen der Aufgabe oder der eigenen Anstrengungsbereitschaft und Leistungsfähigkeit.

Gegensteuern durch mehr Mitarbeiter

- Mitarbeiter zu einem **verspäteten** Softwareprojekt hinzuzufügen, verzögert das Projekt nur noch weiter. (Brooks'sches Gesetz)
 - Das Brooks'sches Gesetz macht nur Aussagen über verspätete Projekte.
 - Ein frühzeitiges (= bevor das Projekt verspätet ist) Einsetzen von Ressourcen verhindert oft Verspätungen
 - Quellen:
 - Einarbeitungsaufwand
 - Projektunruhe
 - Kommunikation
- Interessanter Effekt: **Neto Negative Producing Programmer**
 - Entfernen eines unperformanten, unproduktiven Entwicklers führt zu erhöhter Projektproduktivität

Ständige Anforderungsänderung

- Erschleichung von Funktionalität
 - Ständiges Erweitern der Funktionalität ohne den Projektplan anzupassen
 - Nicht-Informatik-Beispiel: BER
- Erschleichung weiterer Anwendungsbereiche
 - Nicht diskutierte Anwendungsbereiche sind plötzlich sehr wichtig
 - Fehlen eines Bereiches wird als Fehler dargestellt, der dringendst behoben werden muss

Mitarbeiter

Außergewöhnliches

- Einige Projekte funktionieren, obwohl Sie es nach der Lehre nicht sollten. Ein Beispiel von der DFF 2013:
 - Langlebige Software wird seit über 20 Jahren genutzt und gewartet
 - In dem Projekt findet man keine Maßnahme gegen Softwareerosion
 - Das Projekt funktioniert dennoch sehr gut
 - Analysen haben ergeben, dass dies am stabilen Mitarbeiterstamm liegt:
 - 3 Personen sind in Rente gegangen, aber noch ansprechbar
 - Mehrere Personen seit Beginn des Projektes dabei und haben ein vollständiges Hintergrundwissen
 - Der ursprüngliche Architekt der Software ist im Unternehmen und ansprechbar

Problem Mitarbeiter Fluktuation

- Jeder Mitarbeiter, der den Betrieb verlässt
 - Entfernt Hintergrundwissen
 - Entfernt Kompetenz
 - Entfernt Arbeitskraft
- Jeder Mitarbeiter, der in einen Betrieb kommt
 - Ist erst nach der Einarbeitung produktiv
 - Bindet Arbeitskraft
 - Einarbeitung
 - Später: Aufholen von Hintergrundwissen
 - Änderungswünsche an die Projektrichtlinien

Ziele der Mitarbeiterführung

- Herstellen der Identifikation mit dem Produkt
- Optimale Ausbildung
- Vermindern der Störungen
- Mindern des Abwanderungswunsches

Der Mitarbeiter soll motiviert sein, selber aus eigenem Antrieb Softwareerosion zu verhindern (bzw. zu verlangsamen) und erodierte Softwarekomponenten zu restaurieren.

Das kontinuierliche Lernen ist vorrangig zu fördern

- Lernen des Benötigten in direkter zeitlicher Nähe zum Gebrauch
- 70% des ohne Anwendung gelernten gehen verloren
 - Aber ohne Grundwissen kein Lernen in direkter zeitlicher Nähe möglich
- 50% des Wissens ist nach 3 Jahren veraltet (für Informatiker)
 - Aber veraltetes Wissen wird für die Wartung alter Software benötigt
- Vorratswissen ist ineffektive
 - Beispiel Uni:
 - Ihr lernt Wissen aus vielen Bereichen
 - Meist wird nur ca. 1/10tel des Wissens später benötigt
 - Für die meisten ist das benötigte Wissen aber nicht vorhersehbar
 - Ausnahme: **Programmieren** muss jeder Informatiker können!

Herstellen der Identifikation

- Nicht Aufgaben, sondern Kompetenzen delegieren
 - z.B. durch Komponentenverantwortlichkeit
- Am Erfolg teilnehmen lassen
 - Nach erfolgreicher Inbetriebnahme/Update etc.
 - Besuch des Kunden
 - Offizieller Abschluss des Teilprojektes
 - Herausstellen der Kompetenzen
 - Langfristige Bindung an das Produkt
 - Dienstreisen
 - 1-2 pro Jahr

Vermindern der Störungen Mindern des Abwanderungswunsches

- Betriebliche Stabilität
 - Leider heute selten wegen
 - Ad hoc Management
 - Aktienkursgetrieben
 - Kurzfristige Ziele
 - Tendenz in kurzer Zeit ein Denkmal zu setzen
- Arbeitsplatzsicherheit
- Schaffen von Freiräumen
- Vermindern von Multitasking
 - Jeder Bearbeitungswechsel erfordert das Wiedereinarbeiten in eine Aufgabe → Zeitverlust, Stress
 - Aufgabe erst beginnen, wenn alle benötigten Vorleistungen wirklich Vorliegen

Weitere Management Antipattern

Noch ein Meeting mehr wird es lösen

- Verschieben von Entscheidungen auf ein weiteres Meeting
 - Meist gekoppelt mit unzureichender Meetingstruktur
- Besonders schlimm bei verspäteten Projekten
 - Meetings kosten viele Ressourcen
 - 1 h Meeting mit 8 Personen entspricht einem ganzen Arbeitstag!
 - Erhöht die Verspätung
 - Verschiebt eigentliche Arbeiten bis nach der der nächsten Sitzung

Angst vor Erfolg

- verängstigte, defensive Atmosphäre
- Erfolg bleibt aus, weil Aktionen Abgebrochen werden
- Mehr Risiko-Minimierung als Arbeit am eigentlichen Ziel
- Angst durch Erfolge aufzufallen

FLOW
-
Ein Erfolgsfaktor

FLOW (engl. „Fließen“)

- Zustand der vollkommenen und glückbringende Vertiefung in eine Aufgabe
- Kann (leider) nicht erzwungen werden
- Aber man kann die Grundlagen legen:
 1. Bring dich total ein
 2. Konzentriere dich auf den Augenblick
 3. Konzentriere dich auf eine Tätigkeit
 4. Lerne die Arbeit zu genießen
 5. Vermeide Zusammenarbeit mit unzufriedenen Kollegen
 6. Arbeite an Dingen, über die du Kontrolle hast
 7. Strukturiere deine Freizeit

Nächste Woche:
IT Infrastructure Library (ITIL)
–
Grundlagen und Changemanagement