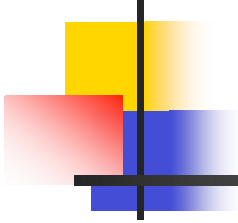


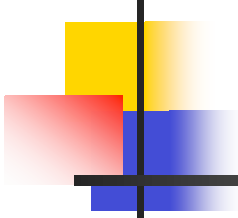
Besprechung Uebung 3

Software Engineering



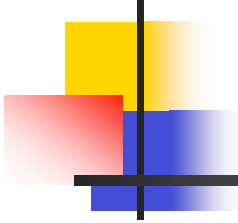
Formelles

- Abgaben:
 - Namen auf Abgaben, Dateinamen ok!
 - Formelle Regeln gelten weiterhin: **1 Punkt Abzug**
- Verschiedenes:
 - Gruppen für die Übungsbearbeitung nicht Einzelpersonen.
 - Eine Abgabe pro Gruppe genügt.
 - Gut gelöst!
- Rechtschreibung, Grammatik
 - Mühsam zu lesen
 - Keine Abzüge
- Strukturierte Argumentation



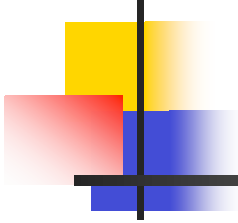
Aufgabe 2.1

- Identifikation der Systemteile/Komponenten
- Generell gut gelöst.
- Zweck:
 - Reduktion der Komplexität
 - Kleinere, besser verstehbare und überschaubare Teillösungen
 - Organisation des Entwicklungsprozesses (Teillieferungen, Ressourcen)
- Komponenten können rekursiv in Komponenten zerlegt werden
- In dieser Aufgabe:
 - Organisation des Prozesses
 - Fokus auf Teilsystemen (→ Teillieferungen)



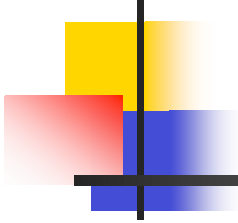
Aufgabe 2.1 (2)

- Korrekturkriterien:
 - Diagramme: Notation (Beschreibung, Legende), Konsistenz
 - Vollständigkeit
 - Trennung von bestehenden Komponenten und neu zu entwickelnden Komponenten
 - Einbindung bestehender Komponenten
 - Terminologie: System, Server, Modul, Service, Tool
- Anwendungsfälle vs. Systemkomponenten
 - Anwendungsfälle: Interaktion aus Benutzersicht
 - Systemkomponenten: Statische Struktur des zu entwickelnden Systems



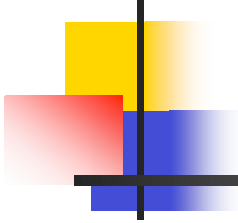
Aufgabe 2.2

- Prozessmodell:
 - Eignung der verschiedenen Prozessmodelle (3 Punkte)
 - Lieferplan für Wachstumsmodell (2 Punkte)
 - Wahl eines geeigneten Prozessmodells (2 Punkte)
- Unterschiedlich gut gelöst!
- Zweck:
 - Komplexität grosser Projekte (grosse Anzahl Beteiligter)
 - Qualität der Lösung



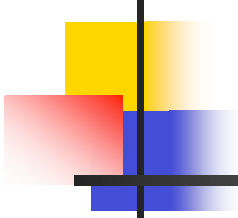
Aufgabe 2.2 (2)

- **Diskutieren** der Eignung von Prozessmodellen:
 - Vorteile
 - Nachteile
 - Auf **konkretes Projekt** (PinkAir) bezogen
- Kriterien für Beurteilung:
 - Grösse des Projekts
 - Risiko
 - Komplexität des Problems / der Lösung
 - Erfahrung der Entwickler in diesem Bereich
 - Problem zerlegbar



Aufgabe 2.2 (3)

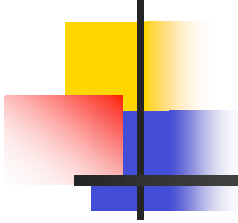
- Ergebnisorientiertes Phasenmodell
 - Projekte nicht zu gross
 - Klar definierbare (nicht klar definierte!) Anforderungen
 - Entwicklungsrisiko eher gering
 - Know-how ist vorhanden
- Wachstumsmodell
 - Grosses System (lange Entwicklungszeit)
 - Volle Funktionalität muss nicht sofort vorhanden sein
 - Vorhaben ist nicht genau definierbar
- Agile Softwareentwicklung
 - Enge Zusammenarbeit mit Kundenvertreter(n)
 - Problem gut zerlegbar
 - Kompetente Entwickler/Architekten



Aufgabe 2.2 (4)

- Kunde hat noch keine genauen Vorstellungen (Fallstudie):
 - Wachstumsmodell
 - Aber: Kunde hat (praktisch) nie implementierbare Anforderungen

→ Anforderungen müssen erhoben werden (Übung 2)
- Erweiterungen werden kommen (insbesondere bei Webapplikationen):
 - Wachstumsmodell
 - Aber: Teillieferungen sind abgeschlossene (prüfbare) Teile
 - Erweiterbarkeit als Qualitätsaspekt (insbesondere für den Entwurf)
- Kombination der Prozesse
 - Wachstumsmodell, Ergebnisorientiertes Phasenmodell für Teillieferung
- Wachstumsmodell nur intern (abhängig vom Kunden)



Aufgabe 2.2 (5)

Agile Software Entwicklung:

Die agile Software Entwicklung versucht die Regeln und Vorgehensweisen für kleine Projekte auf grosse zu übertragen. Ziel ist es die Prozesse möglichst überschaubar und klein zu halten, sodass keine unnötigen Zeitverluste und Überschneidungen entstehen. Dadurch wird auch das Mass der Bürokratie auf ein Minimum reduziert. Zusätzlich sollten die Rückkopplungszyklen kurz sein.

Der Kunde soll aktiv an der Gestaltung des Systems mitwirken und in die Gestaltung miteinbezogen werden, um die Qualität sicher zu stellen.

Aufgrund der kleinen Teilaufgaben, ist es für die Entwickler möglich freier und konzentrierter auf die Teilbereiche einzugehen.

Vorteile:

- Wie beim Wachstumsmodell, aber extremer
- die Arbeit kann von einem kleinen Entwicklerteam bewältigt werden
- kontinuierliche Integration mit geringem Aufwand möglich

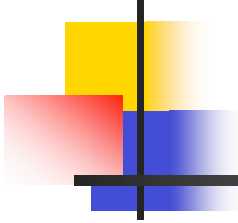
Nachteile:

- Wie beim Wachstumsmodell, aber extremer
- Auftraggeber muss über kompetente Vertreter verfügen, die auch Entscheidungsmacht haben
- kompetenter Softwarearchitekt

Eignung:

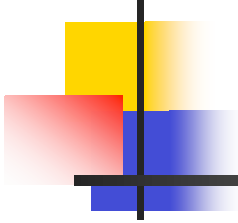
Die agile Software Entwicklung ist ein zweischneidiges Schwert. Hat man kompetente Mitarbeiter und einen geeigneten Auftrag dazu, lohnt es sich nach diesem Modell zu arbeiten. Verliert man jedoch die Kontrolle über das Projekt und führt nicht immer strikte Komponententests durch, versinkt alles im Chaos.

Dieses System bei PinkAir zu implementieren ist etwa aus den selben Gründen wie beim Wachstumsmodell geeignet. Es ist flexibel in der Erweiterung des Systems, es kann über lange Zeit erhalten bleiben und ist überschaubar.



Aufgabe 2.3

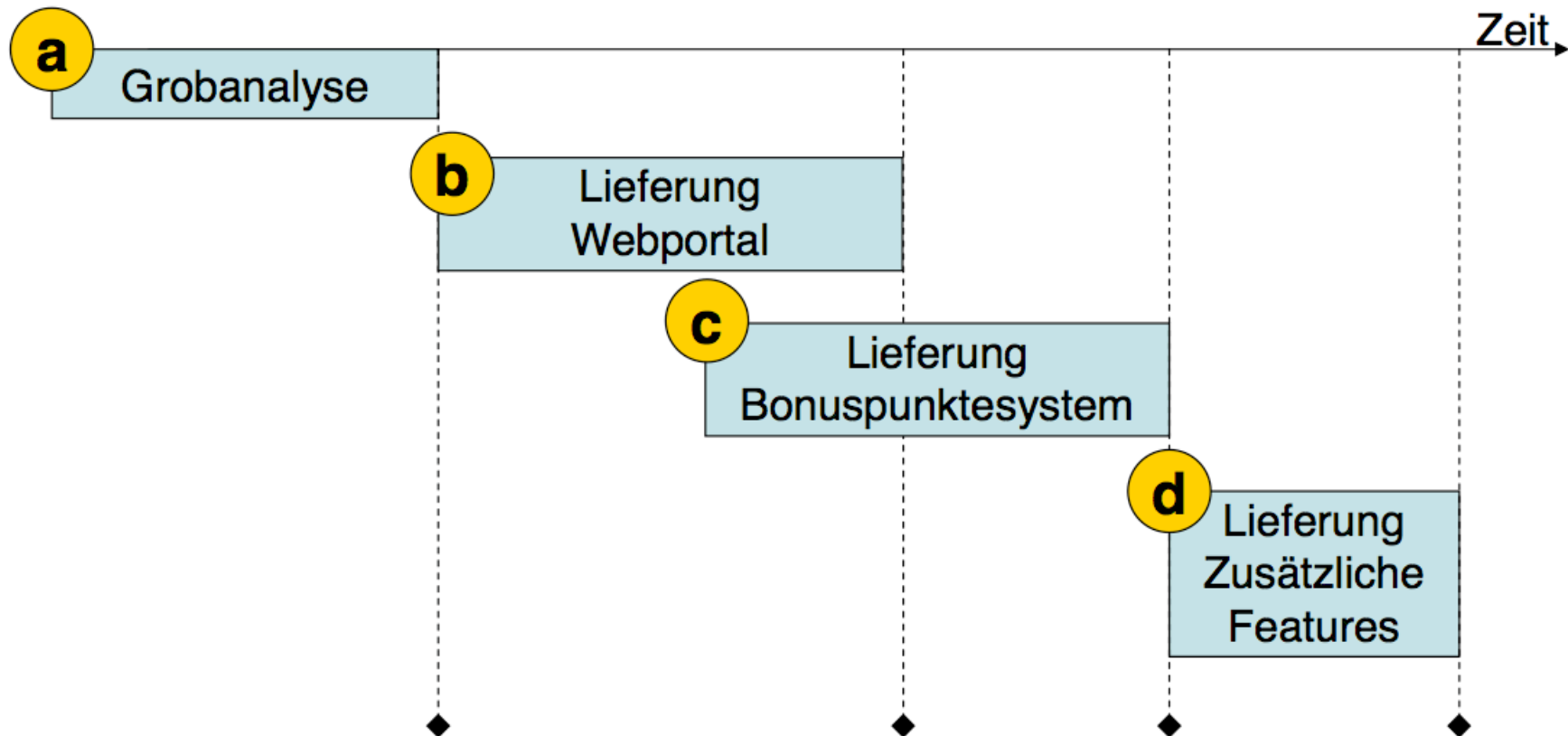
- Meilensteine für das gewählte Prozessmodell
- Gut gelöst!
- Zweck:
 - Prozess strukturieren (Ergebnis, Ressourcen)
 - Fortschritt kontrollieren (Prüfbarkeit der Erreichung)



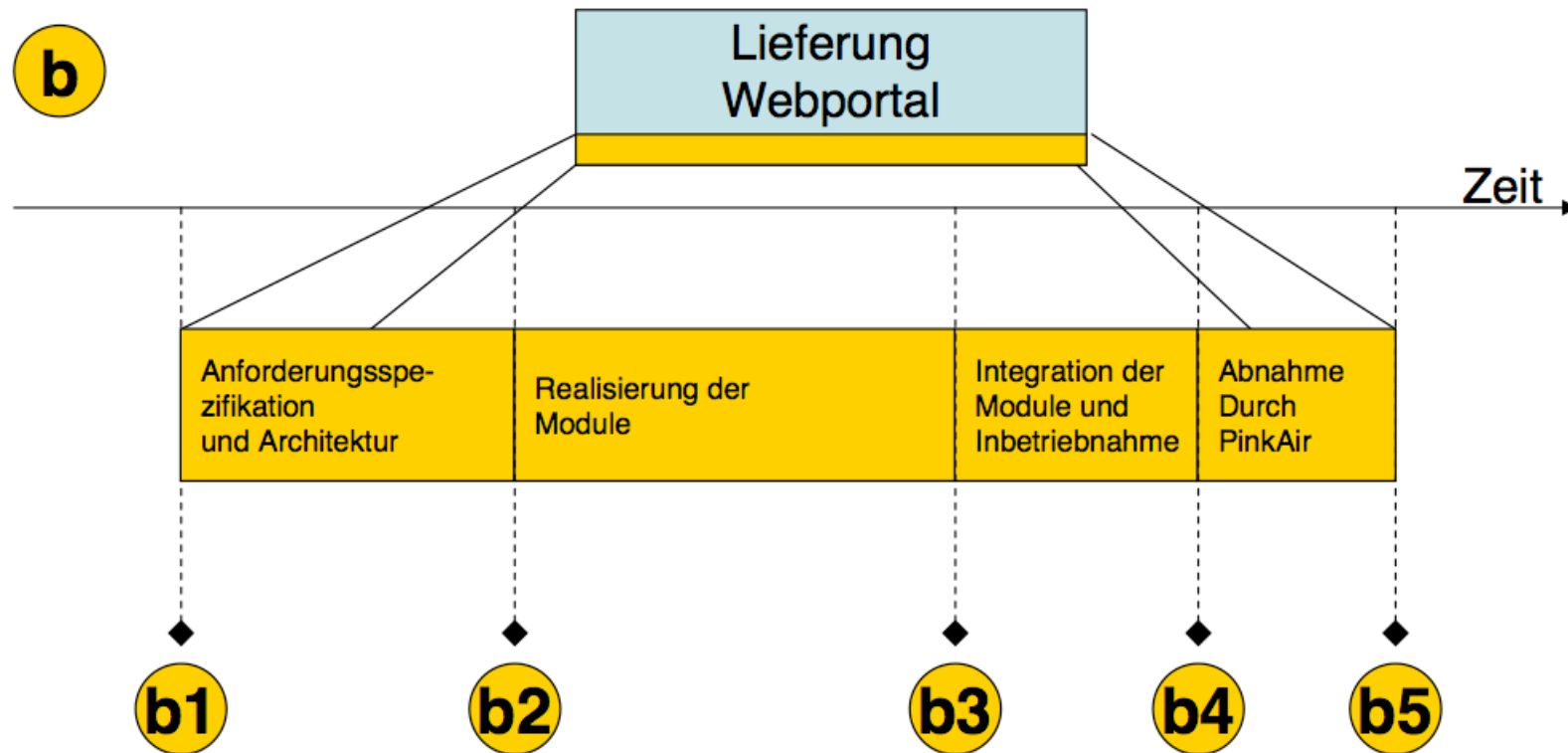
Aufgabe 2.3 (2)

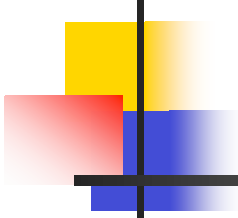
- Korrekturkriterien:
 - Konsistent zu gewähltem Prozessmodell
 - Erreichung prüfbar (Review, Abnahmetest etc.)
 - Wachstumsmodell:
 - Haupt-Meilensteine (Grobanalyse + Iterationen)
 - Neben-Meilensteine (innerhalb einer Iteration)
- Neben-Meilensteine beenden Phasen
- Grobplanung der Lieferschritte
- Integration der Systemteile/Komponenten

Aufgabe 2.3 (3)



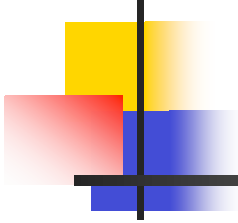
Aufgabe 2.3 (4)





Aufgabe 2.4

- Prototypen in welchen Phasen für welche Komponenten
- Unterschiedlich gelöst!
- Zweck:
 - Frühzeitige Erkennung und Lösung von Problemen
- Exploratives Prototyping
 - Klärung und Bestimmung von Anforderungen
 - Demonstrationsprototyp: Machbarkeit Nützlichkeit
 - Von Benutzern erprobbares, kritisierbares Modell des Systems
- Experimentelles Prototyping
 - Labormuster: Realisierbarkeit kritischer Teile
- Evolutionäres Prototyping
 - Pilotsystem als Kern des zu entwickelnden Systems



Aufgabe 2.4 (2)

- Korrekturkriterien
 - Phase und Komponente angeben
 - Einsatz von Prototypen sinnvoll
 - Art des Prototyps
 - Begründung
- Extreme Standpunkte:
Praktische keine bis zu Prototypen für alle Komponenten, alle Phasen
- Requirements Engineering: Aufwand umgekehrt proportional zum Risiko, das man bereit ist einzugehen.