

Larman chapter 3,4 and 5

- Case Studies
- Inception
- Evolutionary requirements

Case Studies

- **NextGen POS System**

- Kasseapparatsystem, der registrerer salg og håndterer betalinger
- Hardware: computer og stregkode scanner
- Har grænseflade til andre applikationer som fx. skatteberegningssystem og lagerstyringssystem
- Understøtter forskellige brugergrænseflader fx. Web browser terminal, alm. Pc grænseflade baseret på JAVA Swing, PDA, touch screen etc.

- **Matador spil**

- Simulering af spil. En person starter spillet ved at indtaste antal spillere. Herefter observeres spillets forløb.

Hvad dækkes af bogens cases?



Traditionelle forretningsapplikationer består af grænsefladelementer, applikations logik, database adgang og links til ekstern software eller hardware. Bogens eksempler fokuser på applikationslogikken.

Bogens organisering



Iteration 1

Introduces just those analysis and design skills related to iteration one.



Iteration 2

Additional analysis and design skills introduced.

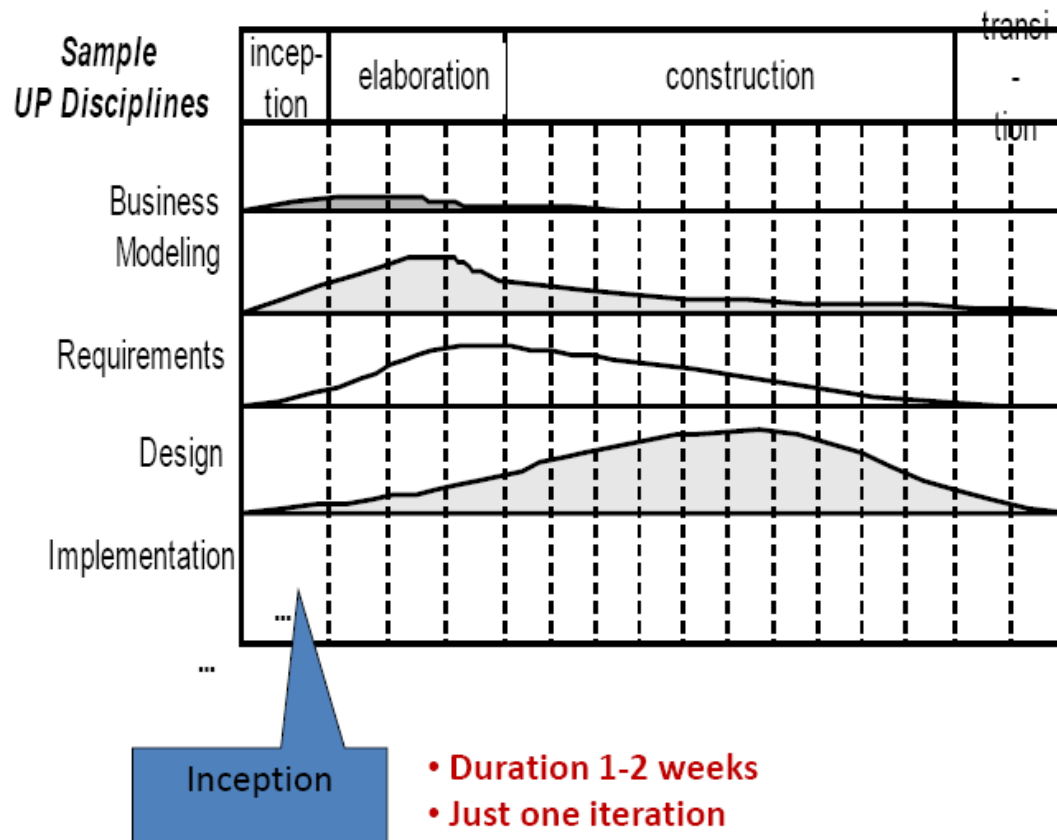


Iteration 3

Likewise.

Inception

Unified process



The relative effort in disciplines shifts across the phases.

This example is suggestive, not literal.

Inception

- Inception, vare typisk relativ kort tid, 1 uge eller nogle få uger. Tit kun en iteration.
- Hvad er vision og business case for projektet
 - Opridser begrundelsen for at indlede projektet
- Er projektet muligt
 - Teknisk, økonomisk, organisatorisk, tidsmæssigt, juridisk
- Købe færdiglavede system eller egenudvikle nyt system

Inception

- Et groft og upræcis skøn over økonomien i projektet
 - Budget ramme
 - Koster 100.000 kr. eller 1. mill. Kr.
- En beslutning om man skal fortsætte eller stoppe projektet
 - Er det værd at investere penge i en fortsat udvikling af systemet
- Har interessenterne (til projektet) en fælles enighed om visionen for projektet

Discipline	Practice	Artifact Iteration→	Incep. I1	Elab. E1..En	Const. C1..Cn	Trans. T1..T2
Business Modeling	agile modeling req. workshop	Domain Model		s		
Requirements	req. workshop vision box exercise dot voting	Use-Case Model	s	r		
		Vision	s	r		
		Supplementary Specification	s	r		
		Glossary	s	r		
Design	agile modeling test-driven dev.	Design Model		s	r	
		SW Architecture Document		s		
		Data Model		s	r	
Implementation	test-driven dev. pair programming continuous integration coding standards	...				
Project Management	agile PM daily Scrum meeting	...				
...						

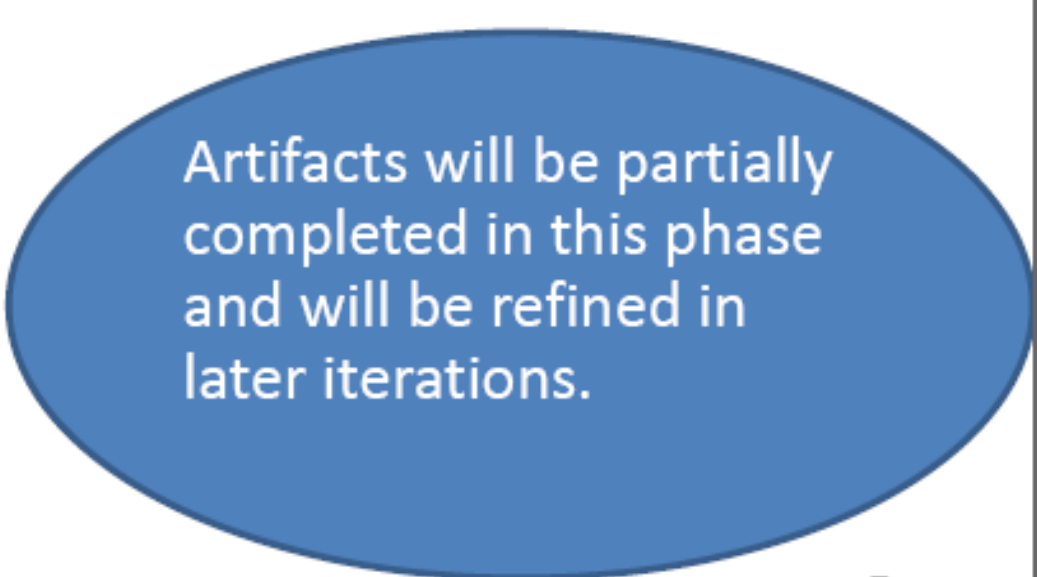
Table 2.1 Sample Development Case. s - start; r - refine

Inception artifacts

- **Vision and business case**
 - Opridser begrundelsen for at indlede projektet
 - Hvad er fordelene ved systemet for virksomheden
 - Hvem ønsker det, og hvorfor?
 - Beskriver de overordnede mål og begrænsninger.
- **Use Case model**
 - Identificerer de fleste af de funktionelle krav
 - Beskriver ca. 10% af de funktionelle krav (høj risiko) i detaljer

Inception artifacts

- **Supplementary specification**
 - Beskriver andre krav, non- funksjonelle krav
- **Glossary**
 - Centrale domæne (domain) terminologi



Artifacts will be partially completed in this phase and will be refined in later iterations.

Other artifacts that may start in inception?

- **Risk list and Risk Management Plan**
 - Beskriver business, tekniske, ressource- og tidsplans risici og ideer til afhjælpning eller håndtering af problemerne.
- **Prototypes og "proof-concepts"**
 - Udspecificer visionen og undersøge tekniske ideer og muligheder
- **Iteration plan**
 - Beskriver, hvad der skal gøres i den første elaboration iteration

Other artifacts that may start in inception?

- **Phase Plan & Software development Plan**
 - Gæt for elaboration fasens varighed. Værktøj, mennesker, uddannelse og andre ressourcer
- **Development Case**
 - Beskrivelse af de tilpassede UP aktiviteter og artifacts (artefakter) til dette projekt.

Create only those artifactst that adds real value to the project!

You misunderstood inception if:

- Inception er mere end en nogle få uger lang
- Du forsøger at definere alle krav
- Skøn og planer forventes at være pålidelige og præcise
- Du definerer arkitekturen (bør gøres iterativt i udarbejdelse)
- Der er ingen business case eller vision artifact (artefakt)
- Alle use cases er skrevet i detaljer
- Ingen af use cases er skrevet i detaljer

Konklusion

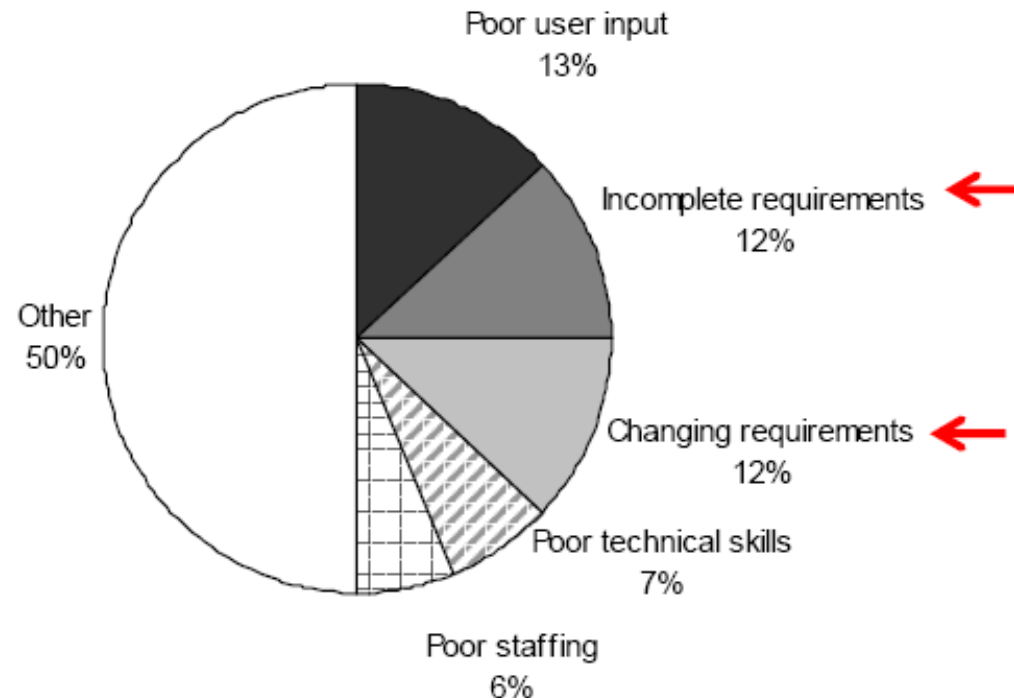
- Den indledende inception fase er ikke meget teknisk.
- Grundlæggende handler den om at beslutte, om det er umagen værd at investere i en dybere udforskning og udvikling af projektet (formålet med elaboration fase)
- Hvis man vælger ikke gå videre er det ikke et negativt resultat ...
 - Hvorfor ikke?

Evolutionary Requirements

Requirements

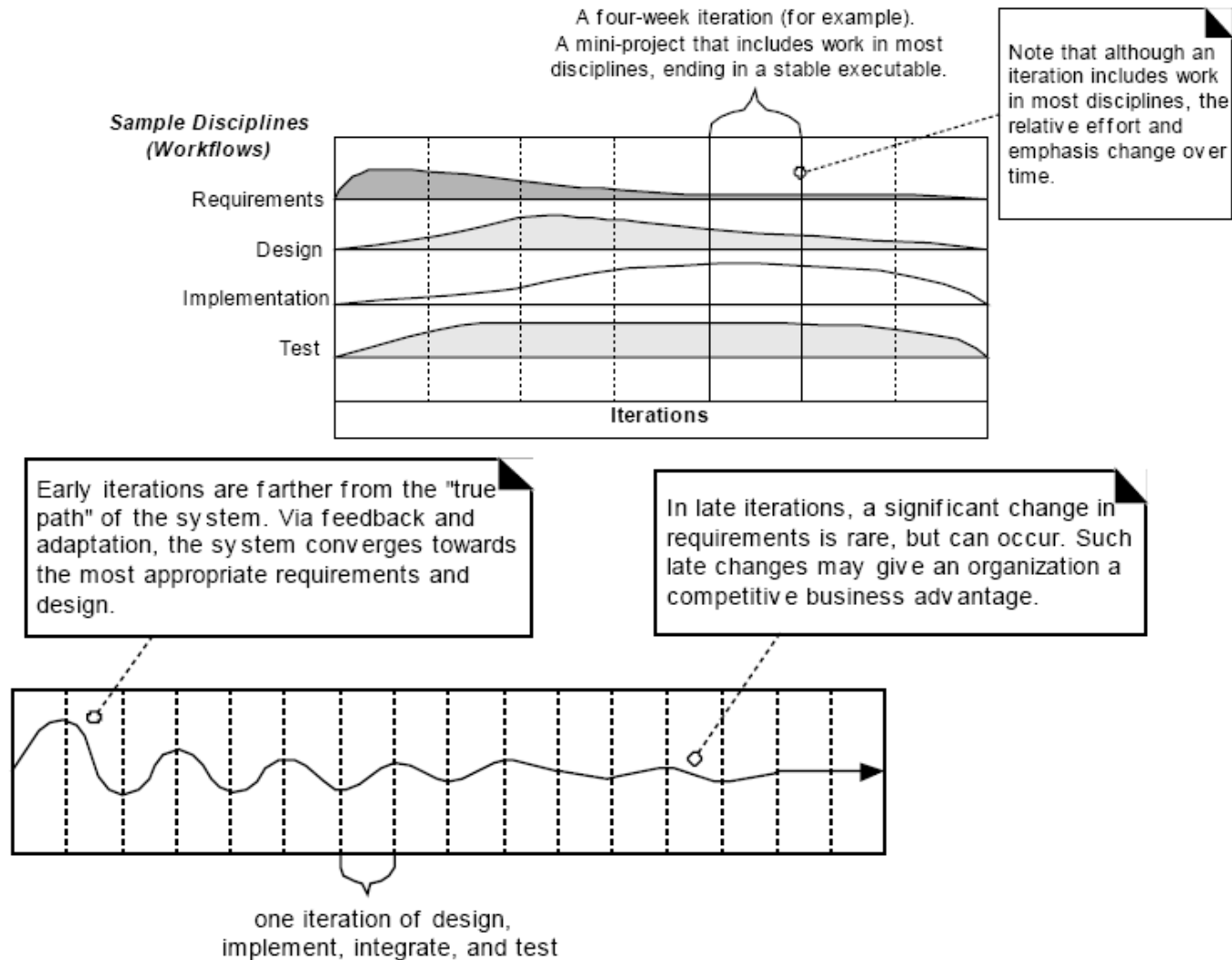
- Håndtering og arbejde med krav er en af de mest risikable dele af Software udvikling
- Men løsningen er ikke at forsøge at specificere og stabilisere alle krav fra begyndelsen (vandfaldsmodel - metode).
- Krav ændringer i løbet af udviklingsprocessen
 - I gennemsnit vil 25% af kravene ændres i løbet af projektforsløbet
- Derfor bør SW udvikles på en iterativ måde med hyppig feedback og tilpasning til forandringer undervejs i udviklingsforsløbet
 - Hver iteration indebærer at man vælge en lille delmængde af kravene, hvor man hurtigt designe, implementere og teste
 - Tidlig feedback (fra brugere, udviklere og prøver) driver udviklingen

Problems in SW systems



The nature of UP supports changing requirements

Iterative refinement of requirements




Øvelse: Læs side 43- 44 i Pressman, og sæt det i forhold til ovennævnte figur

Categories of Requirements

- **FURPS+ model :**
 - **Functionality**
 - **Usability** User interface, help, documentation
 - **Reliability**
 - Frequency of failure, recoverability...
 - **Performance**
 - Response times, resource usage...
- **Supportability**
 - Adaptability, maintainability, internationalization

Categories of Requirements

- **FURPS+ model :**
 - in FURPS+ (everything else...)
 - Design constraints
 - Cost
 - Implementation requirements
 - Languages, hardware
 - Interface requirements
 - Interfacing with externally systems
 - Physical requirements



Use as
Checklist

Alternativ opdeling

- **Functional requirements**
 - Beskriver systemets adfærd og ønskede funktioner
- **Non-functional requirements**
 - beskriver alle andre ønskede egenskaber ved det samlede system
 - Produkt omkostninger, Kapacitet, nuværende og fremtidige
 - Performance (effektivitet, svartid, Opstart), robusthed
 - Tilgængelighed (hvor meget "nede tid" er acceptabelt?)
 - Sikkerhed (data)
 - Sikkerhed (kan undgå skader på mennesker eller miljø?)
 - Dokumentation. Vedligeholdelse mulighederne (udvidelsesmuligheder, genanvendelighed, fejlretning, nye krav)
 - Hvor let skal det være at lave test
 - Support krav, platformskrav

Requirements Artifacts

- **Vision document**
 - Sammenfatter overordnede krav
 - Business case
 - Projektets grundlæggende idé
- **Use-case model**
 - Funktionelle krav beskrevet som et sæt af typiske scenarier omkring anvendelse af systemet
- **Supplementary specifications**
 - Alle andre krav (none-funktionelle)

Requirements Artifacts

- Business rules
 - Domain regler
 - Krav eller regler der overordnede gælder for projektet, som programmet skal følge
 - Typisk eksempel er samfundet skatteregler eller afgift regler såsom moms.
 - Dette punkt **kan** sættes under supplerende specifikationer
- Glossary
 - Definere centrale termer (begreber) i projektet
 - Data dictionary
 - Knytter sig til data, valideringens regler, eventuelle rammer for "acceptable/lovlige" værdier