IMT2243 Systemutvikling

Forelesning 3 :

MODERNE SYSTEMUTVIKLINGSMODELLER

- Oppsummering på de tradisjonelle modellene
- Moderne Systemutviklingsmodeller :
  - Rational Unified Process
  - eXtreme Programming
  - Felles plattform for alle Smidige (Agile) utviklingsmodeller
  - Scrum (forts. i forelesning 4)

Pensum :

Sommerville Kap 2.4 og kap 3,

Kompendium nr 2 (RUP og XP), 3 og 4 (Scrum)

16.01.2013

Ytterligere kommentarer til de tradisjonelle systemutviklings-modellene som er pensum i emnet

Fossefallsmodellen (Waterfall)

Inkrementell Utvikling

- Evolusjonær versjon
- Sekvensiell versjon (inkrementell levering)

Gjenbruksorientert Utvikling

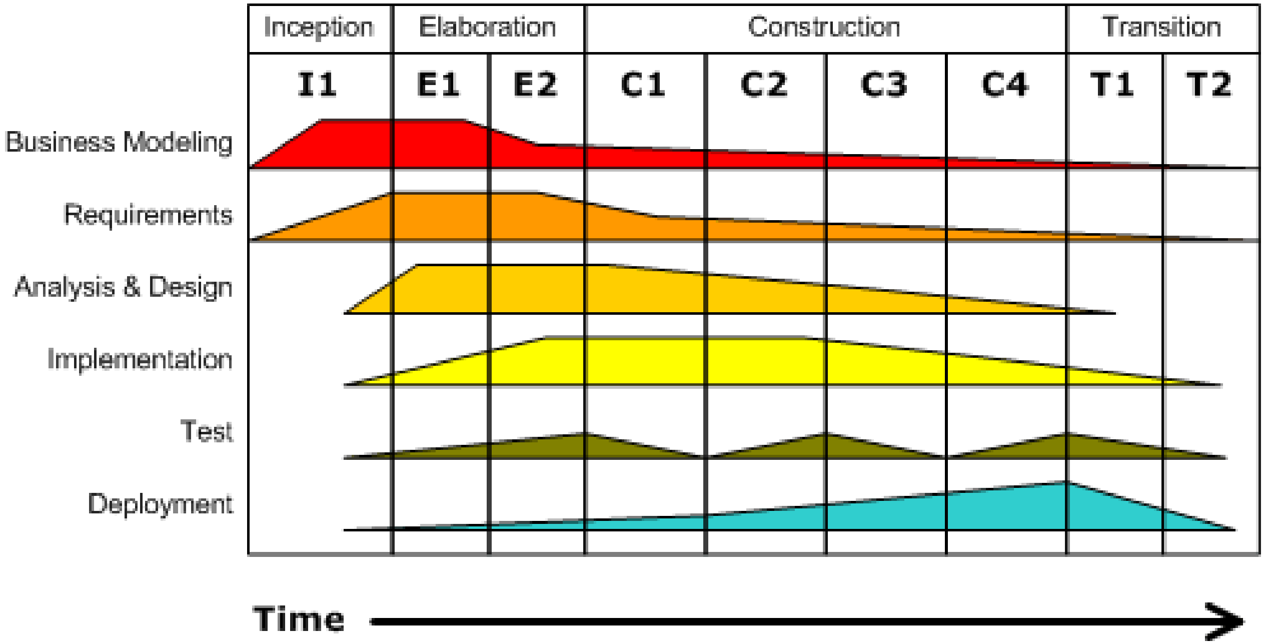
Spiralmodellen

Moderne systemutviklingsmodeller

Rational Unified Process (Heavy model):

Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



Pensum : Sommerville kap 2.4 og Kompedium art. 2. om RUP

Hovedmålet for alle prosjekter der det utvikles ny Programvare :

1 great software development

Pleasing your customer

I used to think *all* programmers got paid in bananas for their projects... but now that I'm developing great software, I get cold, hard cash.

If your customer's unhappy, *everyone's* unhappy!

Every great piece of software starts with a customer's big idea. It's your job as a professional software developer to **bring that idea to life**. But taking a vague idea and turning it into working code—code that **satisfies your customer**—isn't so easy. In this chapter you'll learn how to avoid being a software development casualty by delivering software that is **needed, on time,** and **on budget**. Grab your laptop, and let's set out on the road to shipping great software.

this is a new chapter 1

«Det dreier seg om å utvikle pålitelig, effektiv og brukervennlig programvare som er fleksibel for endringer. Det ligger store utfordringer i å gjøre dette innen realistiske tids- og kostnadsrammer.» (studiebeskrivelse Programvareutviklerne)

I dag er «Time to Market» et stadig mer vektlagt suksesskriterium. Altså at man raskest mulig har fungerende programvare oppe og går (hele eller deler av applikasjonen). Dermed kommer modeller som lever opp til dette mer i fokus.

eXtreme Programming
(Lightweight model) :

Starten : Prosjekt i Chrysler ledet av Kent Beck

(se artikkel under Ressurser, kursorisk dvs ikke pensum)

Verdigrunnlag :

Communication, Simplicity, Feedback, Courage + Respect

12 pratices (opprinnelig, nå utvidet) :

On-site Customer

System Metafor

Små releaser

Planning Game

Par programmering

Felles eierskap

Test før kode

Refactoring

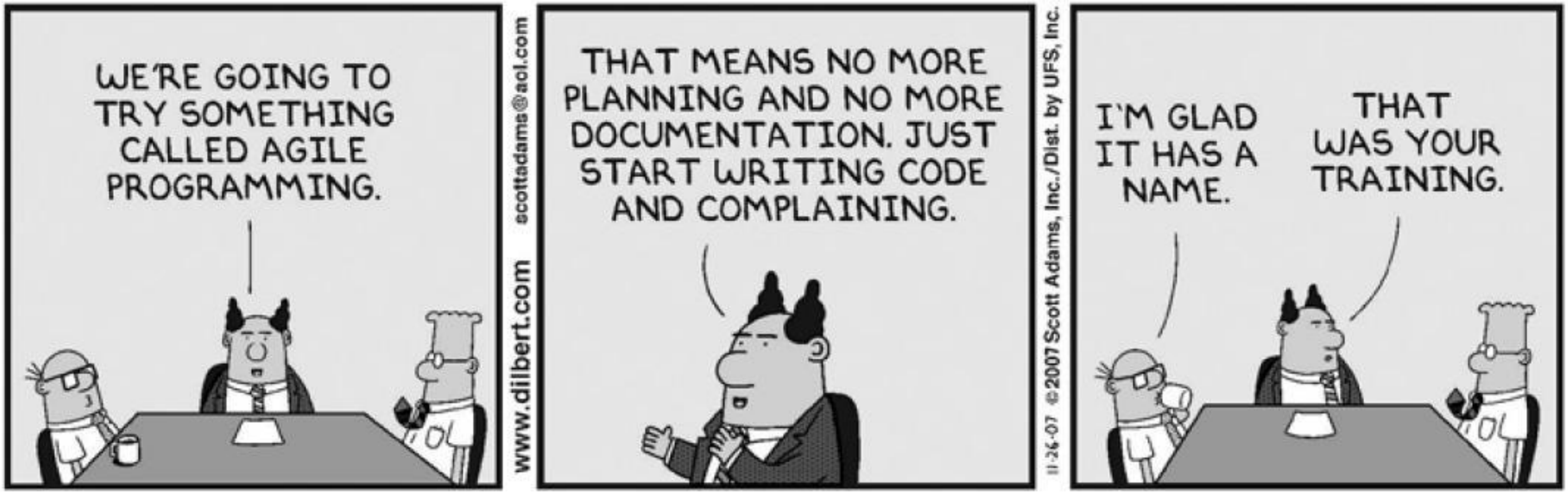
Enkelt Design (DTSTTCPW)

Kontinuerlig integrasjon

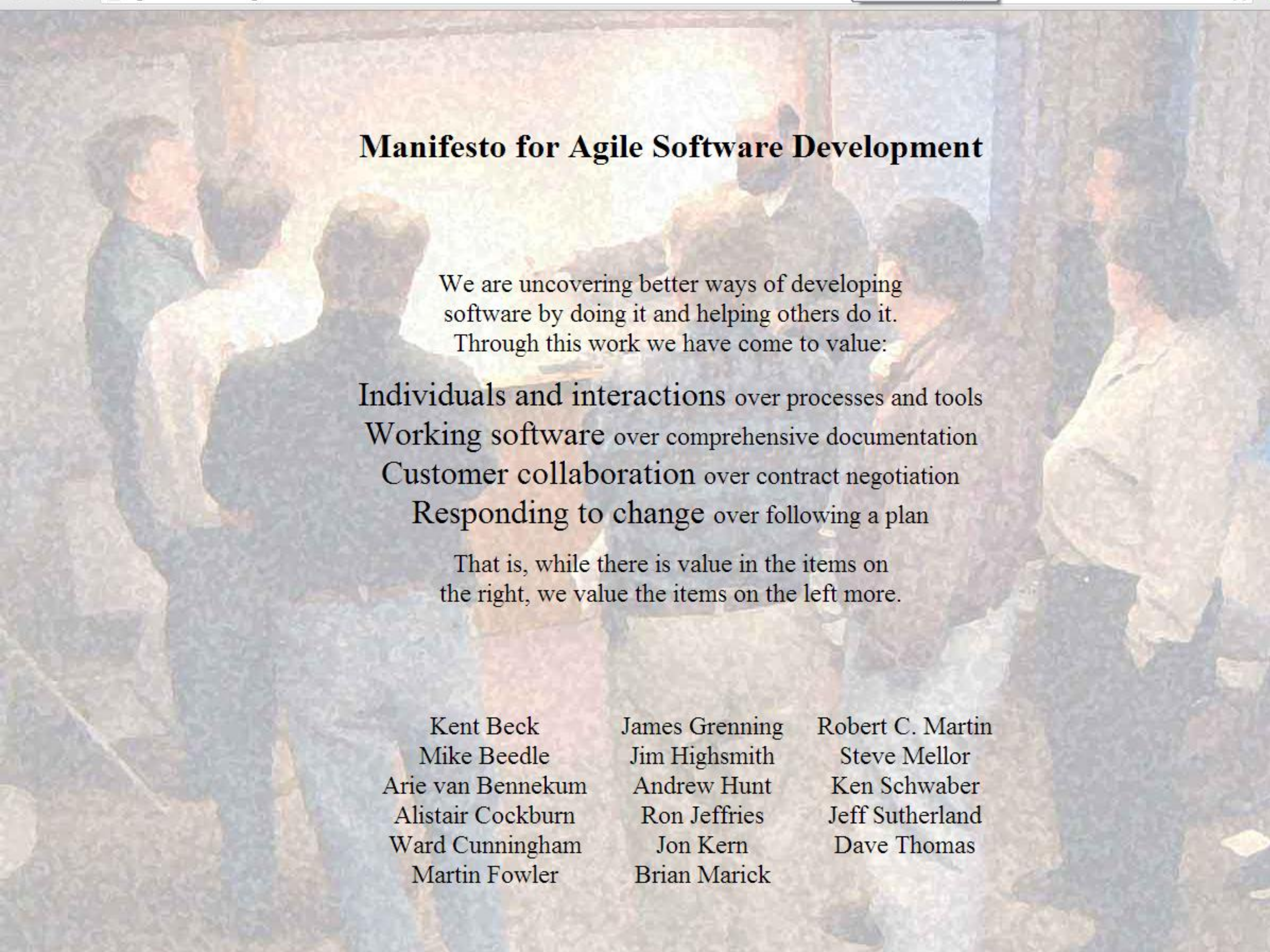
Kodestandard

40 timers uke

Pensum : Sommerville kap 3.3 og Kompedium art. 2. om XP



WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
www.dilbert.com scottadams@aol.com
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
11-26-07 ©2007 Scott Adams, Inc./Dist. by UFS, Inc.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.



Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

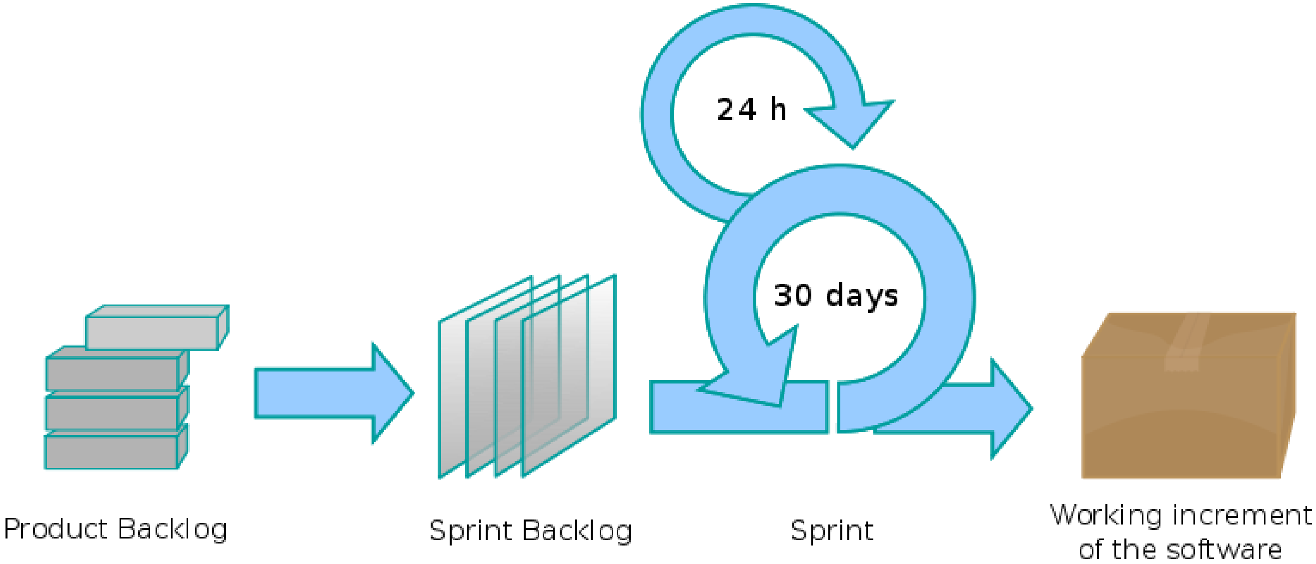
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Noen kilder til
Smidig (Agile) metodikk

- http://agilemanifesto.org/
- http://en.wikipedia.org/wiki/Agile_software_development
- http://search.idg.no/index.cfm?text=smidig&magazine=computerworld

Scrum



Kilde : http://en.wikipedia.org/wiki/Scrum_(development)

Scrum

Scrum Process Mechanics

Roles

PO

Product Owner: set priorities

SM

Scrum Master: manage process, remove blocks

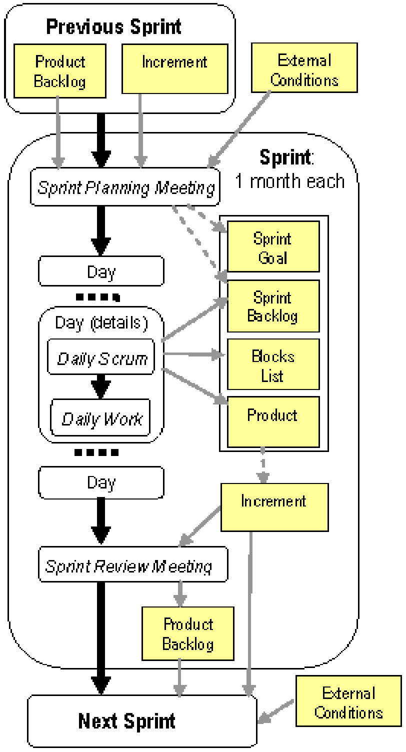
T

Team Members: Develop product

SH

Stakeholders: observe & advise

Sprint Process



Meetings

Sprint Planning Meeting

- SM hosts; all attend. ½ day
- Input: Product Backlog, latest Increment, business and technology conditions
- Output: Sprint Goal, Sprint Backlog
- Agenda:
  1. PO presents Sprint Goal + top-prio Product Backlog items
  2. T estimates items and budget
  3. T selects set of items for sprint according to priorities
  4. T turns business requirements into technical Sprint tasks

Daily Scrum

- SM hosts; all attend; PO observes, catches up on status
- 15 minutes; same time every day
- Each T member (and optionally SM) answers:
  1. What did you do yesterday?
  2. What will you do today?
  3. What's in your way?
- T updates Sprint Backlog
- PO answers T's short questions if asked
- SM updates Blocks List

Sprint Review Meeting

- SM hosts; all attend; ca. 2-3 hours
- Informal, informational
- Agenda:
  1. Demo and discuss Increment
  2. PO formally accepts Product Increment (or not)
  3. SM announces next Sprint Review Meeting

Sprint Retrospective Meeting

- SM hosts; T attends (usually no PO!); ca. 2 hours
- Agenda:
  1. Read prime directive
  2. What went well/wrong during sprint?
  3. How can we improve product, Scrum process, work environment, T performance, engineering practices?

Sprint Cancellation (rare)

- SM calls, if Sprint Goal cannot be met by T
- Ex.: unsolvable impediments; drastic environmental change; severe misestimation.

Artifacts

Product Backlog

- List of business requirements & issues
- Owned by PO
- Anybody can add to it, but only PO prioritizes

Sprint Goal

- One-sentence summary, declared by PO
- Mutually acceptable to T and PO

Sprint Backlog

- List of technical tasks per Product Backlog Item
- Owned by T, status & estimates updated daily
- Only T modifies it (PO must not change scope!)

Blocks List

- Impediments, blocks, pending decisions
- Owned by SM, updated daily

Product Burndown Chart

- Visualizes overall progress (estimated remaining efforts) and team velocity
- Updated after each sprint

Sprint Burndown Chart

- Visualizes sprint progress (estimated remaining time)
- Helps detect problems in sprint
- Updated daily

Definition of Done (DoD)

- List of quality criteria applicable to all requirements
- Mutually accepted by T and PO

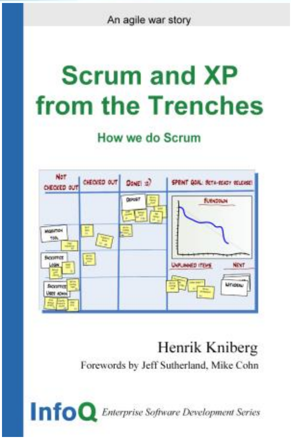
Product Increment

- Potentially shippable product version
- DoD-compliant (tested, documented etc.)
- Delivered once per sprint

Information Radiators

- Scrum and other artifacts (e.g. architecture diagrams)
- Easily accessible for SH and other interested parties
- Big posters/monitors in public areas or project wiki
- Purpose: provide max. transparency about project

Product Backlog


An agile war story

Scrum and XP from the Trenches

How we do Scrum

Henrik Kniberg

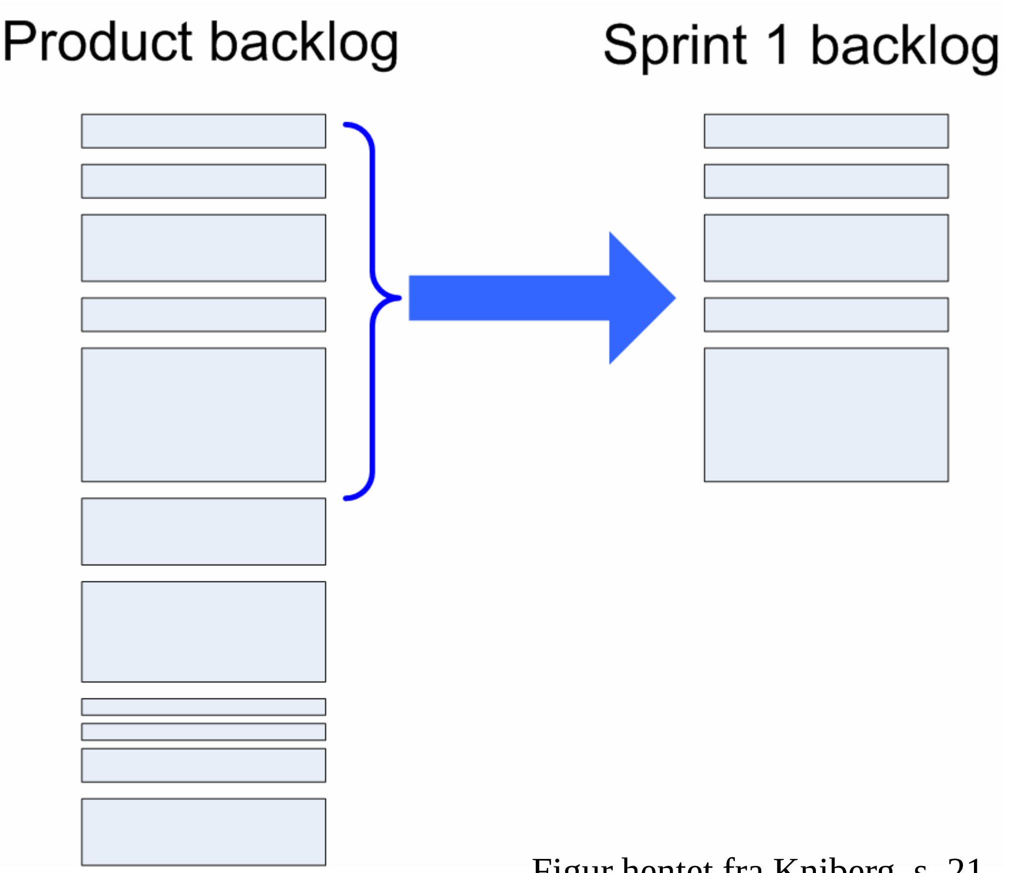
Forewords by Jeff Sutherland, Mike Cohn

InfoQ Enterprise Software Development Series

ID	NAME	Import-ance	Esti-mate	How to demo	Notes
1	Deposit	30	5	Log in, Open dep page, dep $10, go to balance page and check that it has increased	Need UML seq. Diag. No encrypt.
2	See your own transact history	10	8	Log in, click on “trans”.Do deposit. Go back to trans, check taht the new deposit shows up.	Use pageing to aviod large DB-queries..
..					

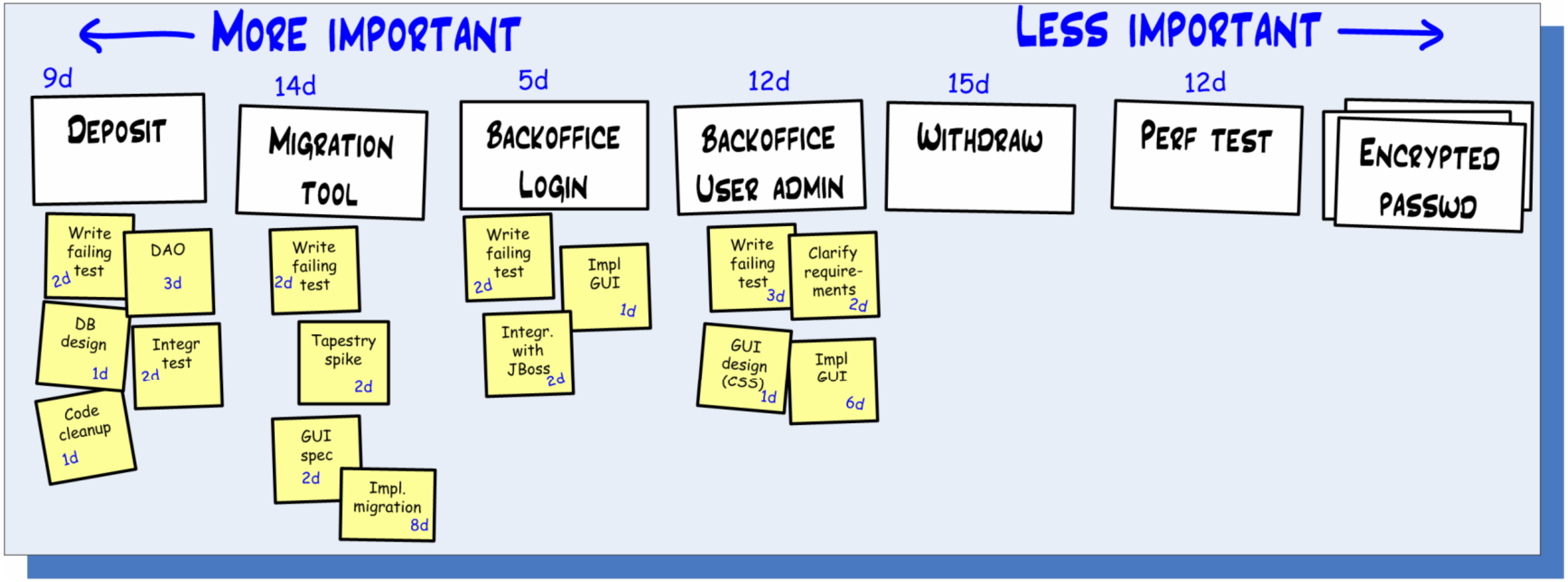
Figur hentet fra Kniberg, s. 10

Overgang Planlegging -> Sprint



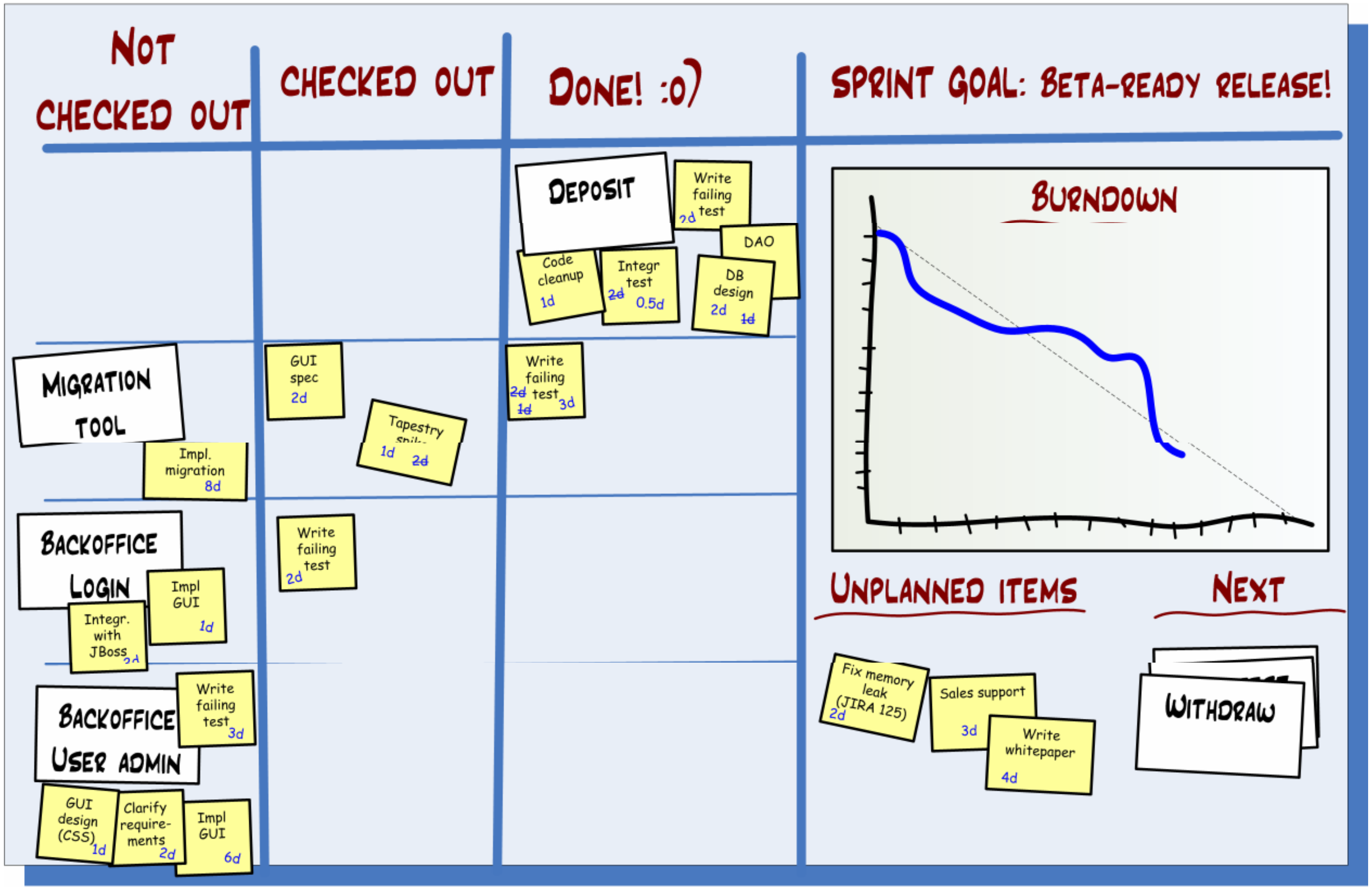
Figur hentet fra Kniberg, s. 21

Estimeringmetodikk i Scrum



Figur hentet fra Kniberg, s. 31

Taskboard – styringsredskap under sprinten



Figur hentet fra Kniberg, s. 49