

Projektstyring i softwareudvikling

Udklip fra bøgerne
“Software Development” af Head First
“Digitalt Design og Udvikling” af Systime

Teknikfaget Digital Design og Udvikling, HCØ Lyngby

Projektstyring hvad og hvorfor?

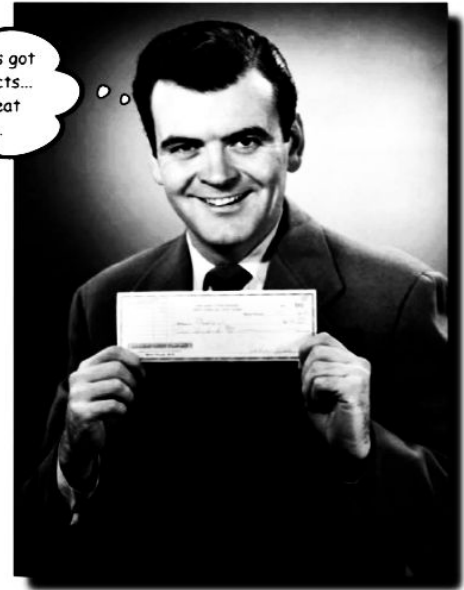
At tænke over hvad, hvem og hvornår?
Både før, imens og efter.

Fordele:

- *Færre fejl*
- *Når deadlines*
- *Man lover ikke for meget*

Chapter 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.



PAPER PROTOTYPE

En papmodel af produktet er primitiv
Men også billig og effektiv til at afprøve en ide

Low-fidelity - fra “Digitalt design og udvikling, Systime”

Kun fantasien sætter grænser for, hvilke slags materialer som kan anvendes til udarbejdelsen af en low-fi-prototype. Det handler om at give projektgruppen frihed til at konstruere præcis, hvad de har brug for.

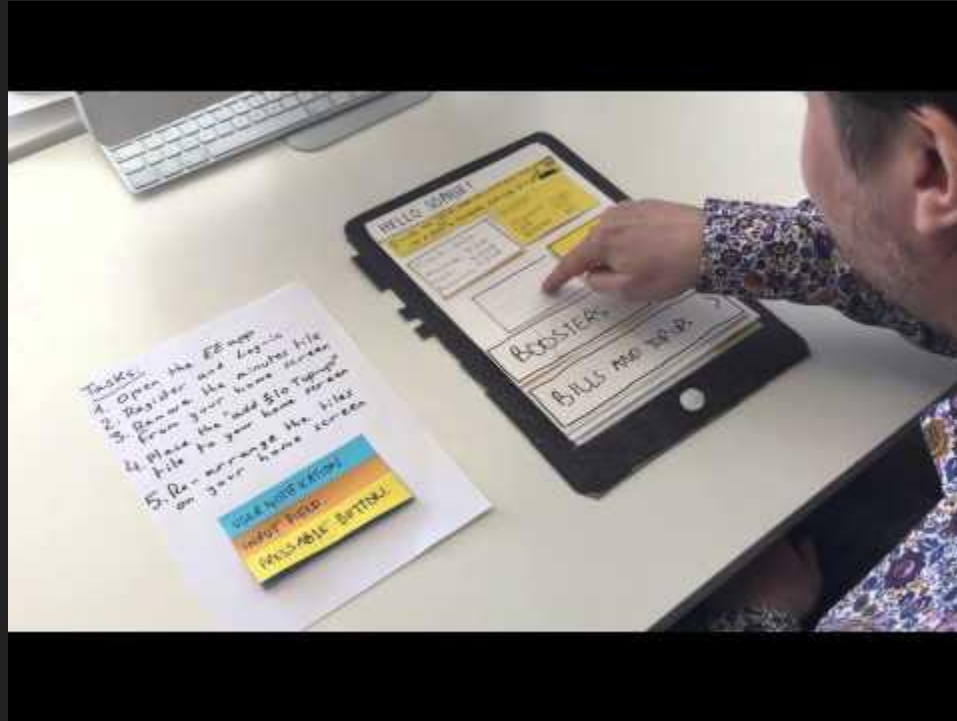
Her er en liste med nogle eksempler på, hvad en værktøjskasse til udvikling af low-fi-prototyper kan indeholde:

- Forskelligt farvet pap og papir
- Lim/tape
- Blyanter, kuglepenne og markers
- Farveblyanter og tuscher
- Sakse
- Lego/Duplo
- Træpinde i forskellige størrelser
- Snor
- Terninger
- Meeples (spillebrikker)
- Perler

En low-fi-prototype er den mest lavpraktiske udgave af produktet, som projektgruppen kommer til at udarbejde i løbet af et projekt. Den har kun til formål at konceptualisere projektgruppens idé, så projektgruppens medlemmer har nogenlunde samme forståelse for produktets funktionalitet og udformning. Efter at have udarbejdet en low-fi-prototype kan projektgruppen fortsætte udviklingen af produktet med en fælles vision for produktet.



“Low fidelity prototype testing of the EE app”, fra youtube



ANVENDELSE AF “PAPER PROTOTYPE”

Billig måde at afprøve et konceptet.

Her et par gode råd:

- Gør jer umage med at bygge en god model
 - Så vil alle i gruppen have det samme mentale billede af produktet
- Afprøv modellen på en testperson, og optag det på video
 - Giver muligheden for at vurdere produktets brugbarhed
- Afhængigt af testresultater justeres “krav” og “user stories”
 - Der er mange forskellige måder skaffe testdata, f.eks. interview

USER STORIES

Inden projektet begynder skal man have styr på de overordnede krav til funktionalitet og brugerflade

Det kan gøres med user-stories

User story, fra “Software Development, Head First”:

Your requirements must be CUSTOMER-oriented

A great requirement is actually written **from your customer's perspective** describing what the software is going to do **for the customer**. Any requirements that your customer doesn't understand are an immediate red flag, since they're not things that the customer could have possibly asked for.

A requirement should be written in the customer's language and read like a **user story**: a story about how their users interact with the software you're building. When deciding if you have good requirements or not, judge each one against the following criteria:

User stories SHOULD...

- ☐ ... describe **one thing** that the software needs to do for the customer. ← Think “by the customer, for the customer”
- ☐ ... be written using language that **the customer understands**. ←
- ☐ ... be **written by the customer**. ← This means the customer drives each one, no matter who scribbles on a notecard.
- ☐ ... be **short**. Aim for no more than three sentences.

You should be able to check each box for each of your user stories.

User stories SHOULD NOT...

- ☐ ... be a long essay. ←
- ☐ ... use technical terms that are unfamiliar to the customer.
- ☐ ... mention specific technologies.

If a user story is long, you should try and break it up into multiple smaller user stories (see page 54 for tips).

Title: Choose seating.....

Description: A user will choose aisle or window seating, be able to select the seat they want, and change that seat up to 24 hours before the flight.

Title: Choose aisle/window seat.....

Description: A user can choose either aisle or window seating.

Title: Choose specific seat.....

Description: A user can choose the actual seat that they want for a shuttle flight.

Title: Change seating.....

Description: A user can change their seat up to 24 hours before launch, provided other seat options are available.

Title: Order in-flight meals.....

Description: A user will choose which meal option they want, from a choice of three, and be able to indicate if they are vegetarian or vegan.

Title: Select from meal options.....

Description: A user can choose the meal they want from a set of three meal options.

Title: Specify vegetarian meal.....

Description: A user will be able to indicate that they are vegetarian when selecting their meal options.

Title: Specify vegan meal.....

Description: A user will be able to indicate that they are vegan when selecting their meal options.

ANVENDELSE AF USER STORIES:

- User Story:
 - **Ikke teknisk beskrivelse:** Beskrivelse af krav og funktionalitet
 - **Prioriteret:** Det vigtigste (måske sværeste) skal laves først
 - **Tidsestimat:** ...

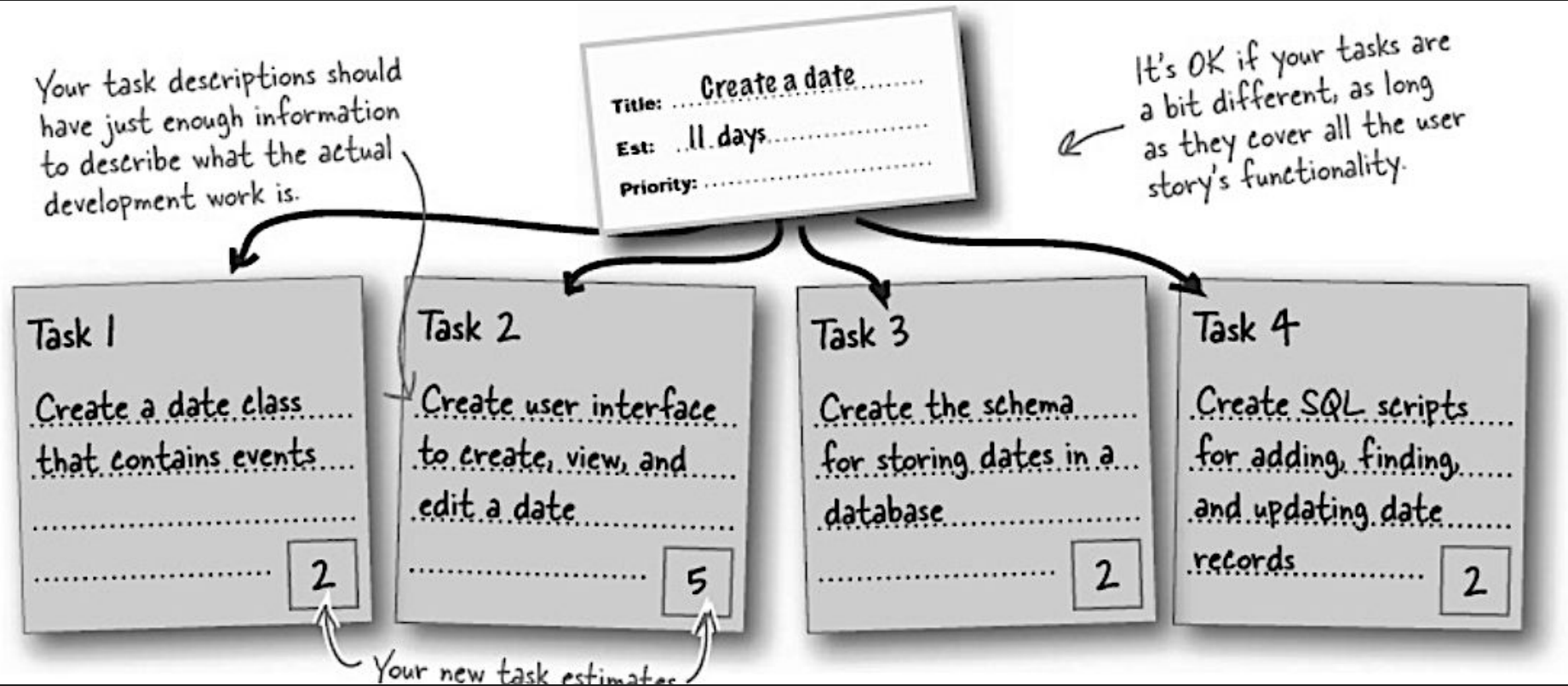
Specielt for teknikfaget digital design og udvikling:

Der kan være fordele ved at tidsestimere flere gange, hvis man arbejder med rigtige kunder. Men i teknikfaget har vi hverken særligt meget tid eller kunder så vi foreslår i venter med at tidsestimere til i har en bedre ide om hvordan programmet udarbejdes rent teknisk

TASKS

Design planen brydes ned i små delopgaver på nogle timer -
disse delopgaver kaldes tasks

Tasks, fra “Software Development, Head First”:



ANVENDELSE AF TASKS

Hver user story skal deles op i **task's**.

Alle **task's** kaldes projektets **backlog** og er det samlede arbejde der skal gøres.

En task bør indeholde:

- Titel Titlen er vigtig når projektgruppen snakker sammen
- User story Hvilken user story stammer den fra
- Beskrivelse Hvad skal der rent teknisk skal laves
- Tidsestimat Det er meget svært at regne ud hvor langt tid noget tager og det er også afhængigt af programmøren men det skal gøres!
- Prioritet

ITERATIONER

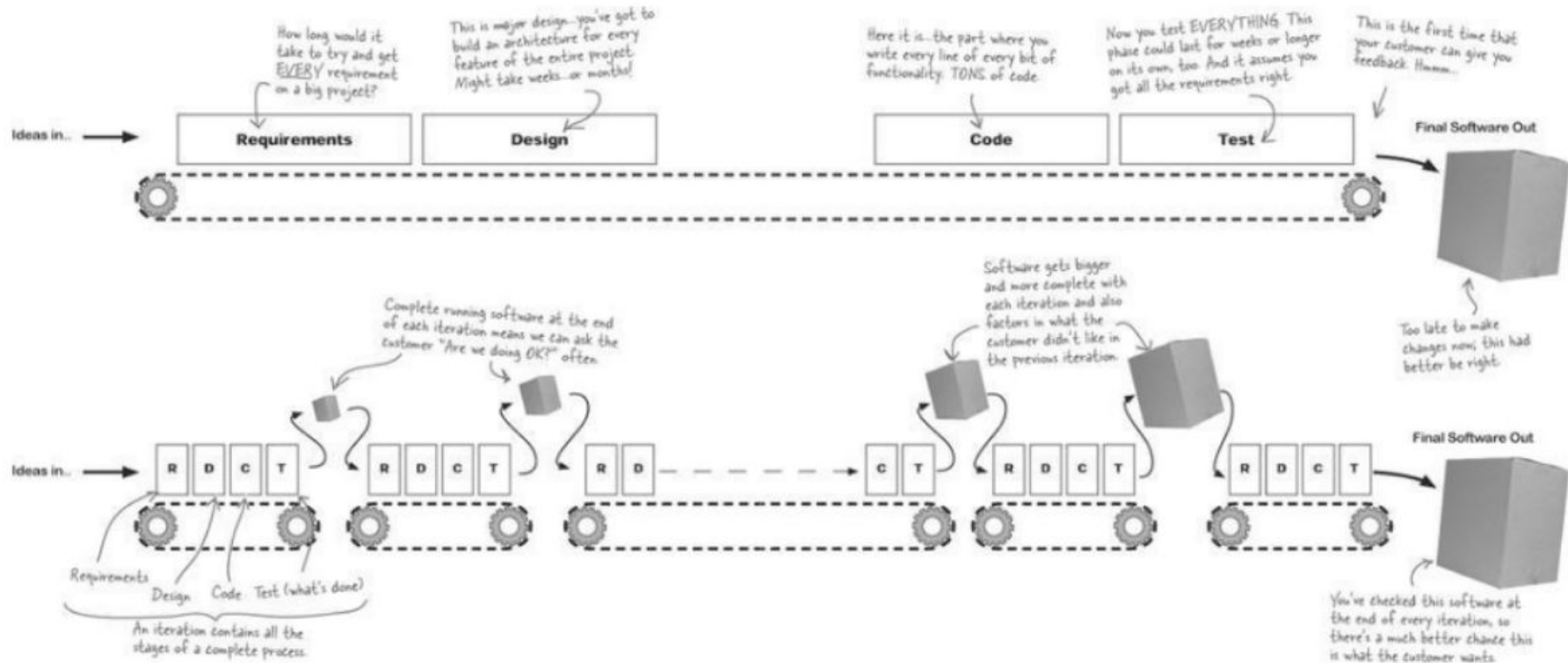
Med en iterativ projektstyring kan du forhindre mange fejltagelser

Iteration, fra “Software Development, Head first”:

Each iteration is a mini-project

With iteration, you take the steps you'd follow to build the entire project, and put those steps into **each iteration**. In fact, each iteration is a mini-project, with its own requirements, design, coding, testing, etc., built right in. So you're not showing your customer junk... you're showing them well-developed bits of the final software.

Think about how most software is developed: You gather requirements (what your customer wants), build a design for the entire project, code for a long time, and then test everything. It looks a bit like this:

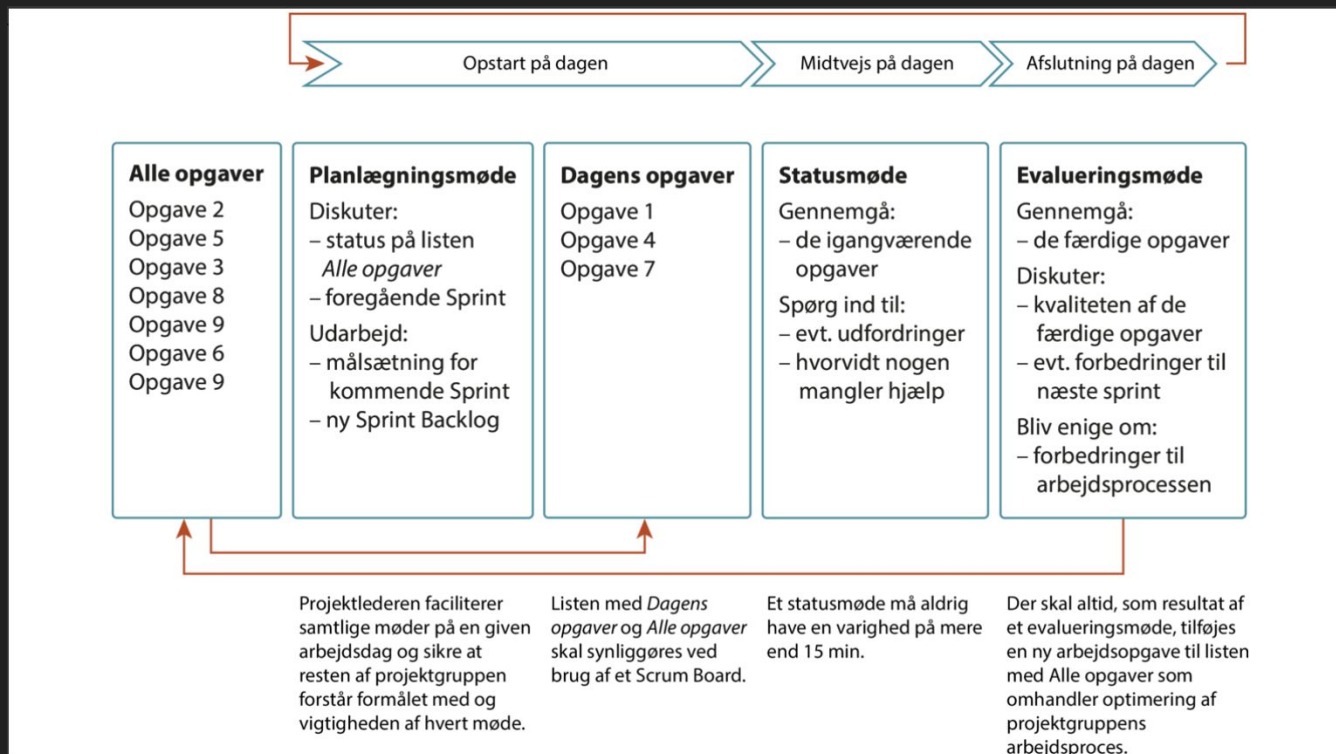


Fordel jeres user stories og tasks

Iteration 1			Total Days of work: 42
<p>Title: View shuttle deals</p> <p>Est: 12 days</p> <p>Priority: 10</p>	<p>Title: Book a shuttle</p> <p>Est: 15 days</p> <p>Priority: 10</p>	<p>Title: Pay with Visa/MC/PayPal</p> <p>Est: 15 days</p> <p>Priority: 10</p>	
Iteration 2			Total Days of work: 38
<p>Title: Manage special offers</p> <p>Est: 13 days</p> <p>Priority: 10</p>	<p>Title: Choose seating</p> <p>Est: 12 days</p> <p>Priority: 20</p>	<p>Title: Order In-flight meals</p> <p>Est: 13 days</p> <p>Priority: 20</p>	
Iteration 3			Total Days of work: 39
<p>Title: Review flight</p> <p>Est: 13 days</p> <p>Priority: 30</p>	<p>Title: View flight reviews</p> <p>Est: 12 days</p> <p>Priority: 30</p>	<p>Title: Apply for Space Miles Loyalty Card</p> <p>Est: 14 days</p> <p>Priority: 40</p>	

Fra bogen “Digitalt, design og udvikling et teknikfag - Systime”

Dette er et forslag til hvordan man strukturerer arbejdet indenfor én iteration



Læg mærke til at de mange iterationer et langt udviklings-projekt kan indeholde

Læg mærke til at de mange iterationer et langt udviklings-projekt kan indeholde

Iterationer	Papir	Adobe XD		Færdigt produkt		
Sprints	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6
Milepæle						
Deadline	Dato	Dato	Dato	Dato	Dato	Dato

Iterationer	Papir	Adobe XD		Færdigt produkt		
Sprints	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6
Milepæle	Analyse	Analyse	Design	Analyse	Produktion	Produktion
	-målgruppe	-intern test	-Adobe XD	-test målgruppe	- funktioner	-funktioner
	-kontekst				-UI	-UI
		Design	Undersøgelse	Design		
	Idegenerering	- funktioner	-test målgruppe	-evt. redesign pga. test		Undersøgelse
	-nye	- UI				-test målgruppe
			- farver	Produktion	Produktion	
	Koncept		-forberedelse til sprint 4-6	-funktioner		Afsluttende evaluering
	-skitsering	Produktion		-UI		
		- UI elementer				
	Undersøgelse					
	-intern test					
Deadline	1/8	2/8	3/8	4/8	5/8	6/8

ANVEND ITERATIONER

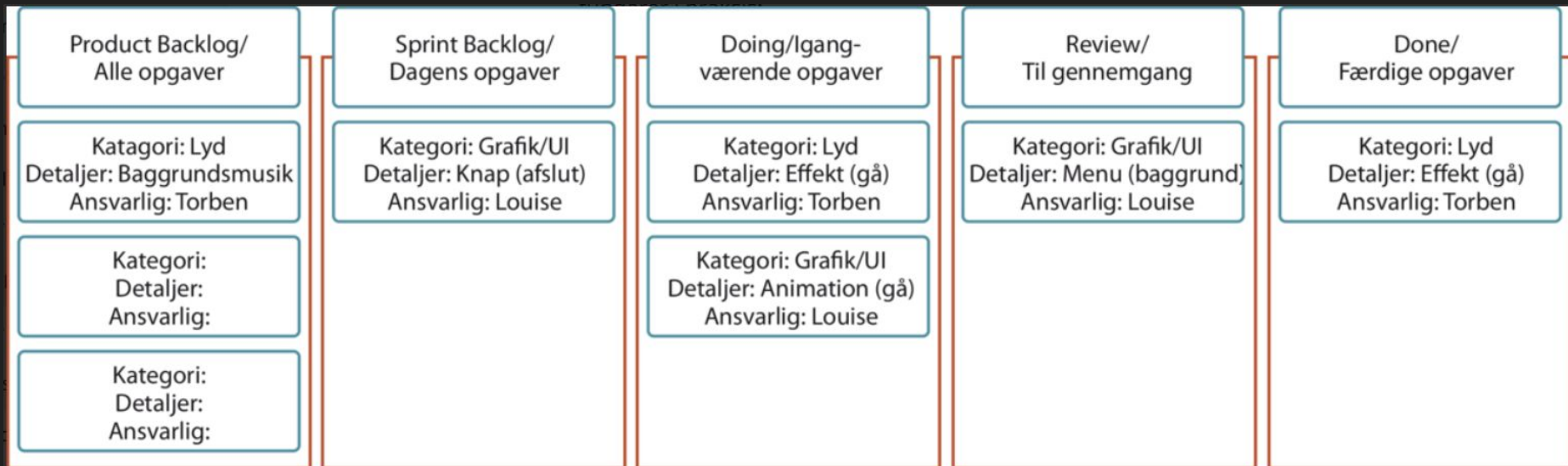
- Et “miniprojekt”. Et helt projektet består af flere iterationer
- Produktet bør være brugbart efter hver iteration!
- Før iterationen:
 - Plan: Hvad der skal laves? (hvilke tasks)
- Efter iterationen:
 - Status: *Hastigheden? Mangler? Hvad virker?*
 - Evaluering: *Hvorfor? Hvordan kan det gøres bedre?*
 - Brugertest: *Brugertest kan evt. anvendes*
 - Justering: *Ændring af tidsestimater*
Tilføj krav
Evt. fjernes nogle krav - snak med “kunden”

Project board

Nu skal der programmeres - første iteration påbegyndes
Nu skal vi lave et “project board” så vi kan overskue progressionen

Scrum-board fra bogen “ddu - systime” (kan også kaldes Kanban-board)

Lav eventuelt kolonne:
“To be tested”



ANVENDELSE AF PROJECT BOARD

For hver iteration bør projektgruppen optimalt have et “project board” med minimum fem kolonner:

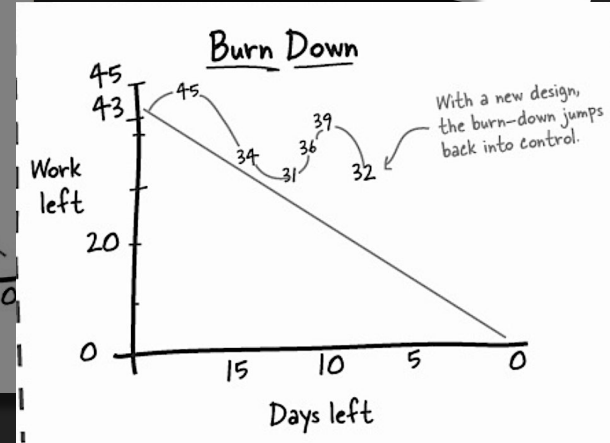
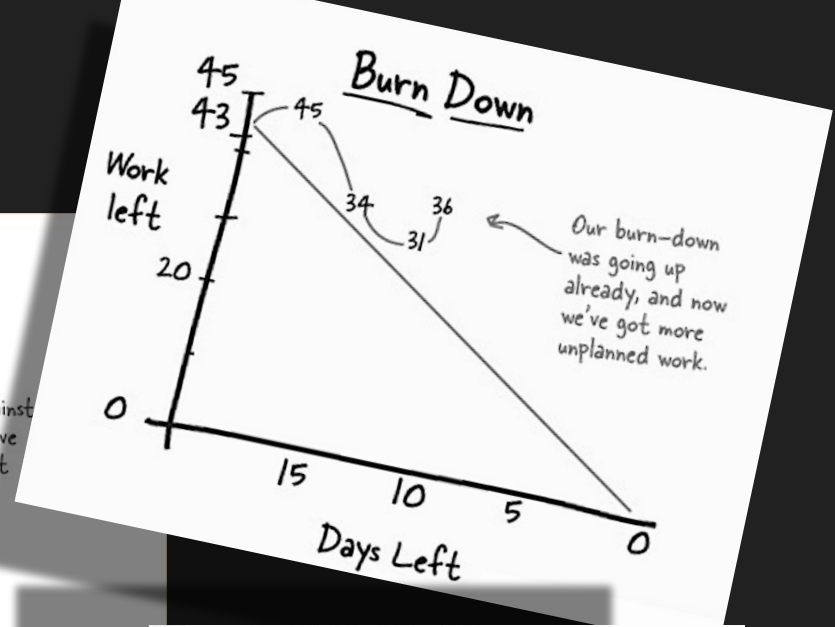
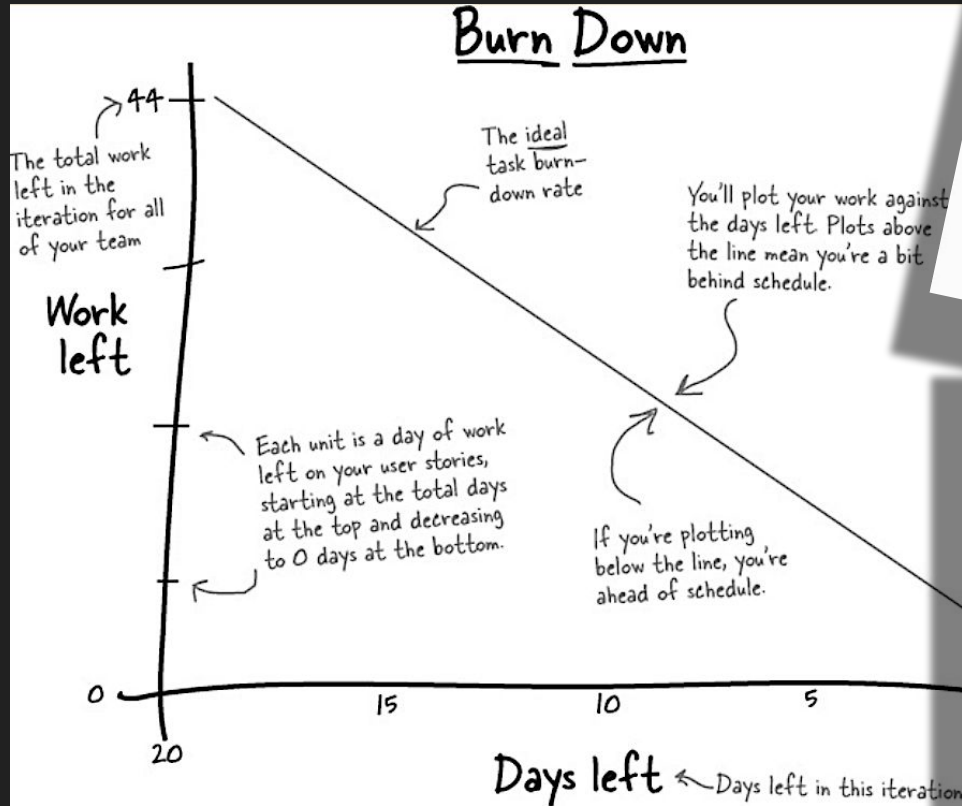
- to do: tasks der ikke er påbegyndt
- in progress: tasks der er under udvikling
- done: udvikleren er færdig med kodningen
- in review: en anden i gruppen tester for at sikre at alt virker
- completed: Tasks der er blevet godkendt af testeren

Burn Down Chart

Udviklingshastigheden vurderes vha. et burn down chart

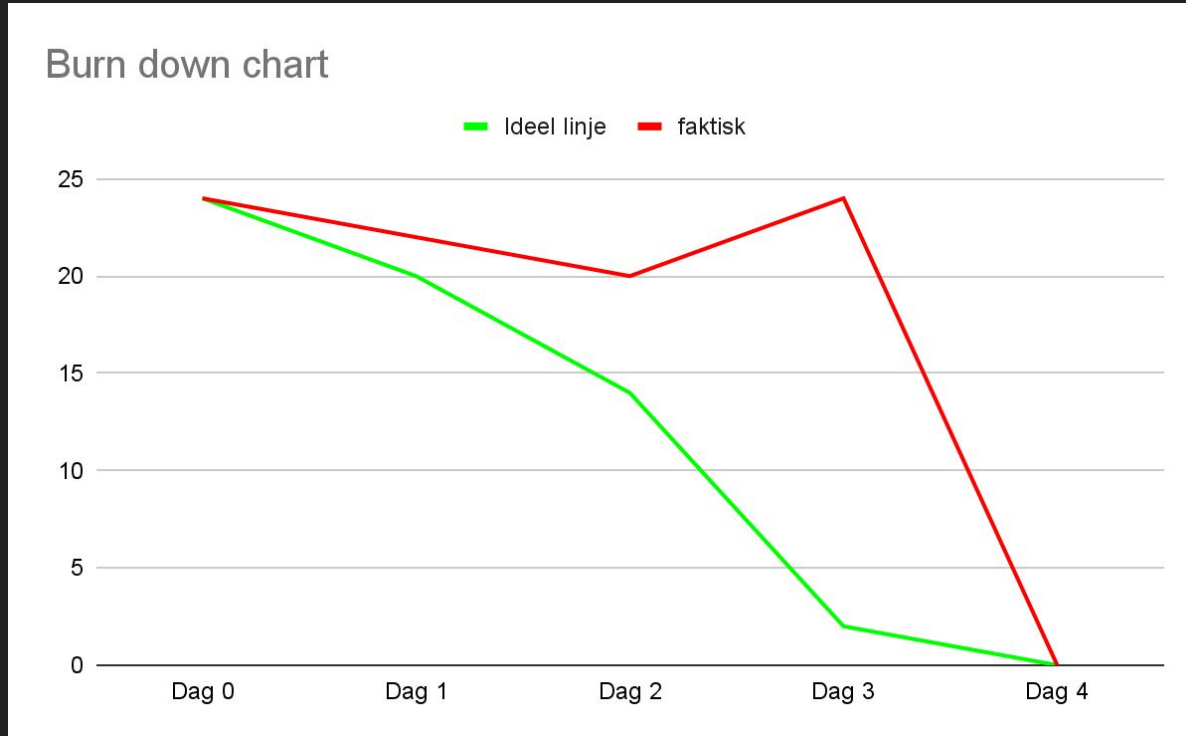
“burn down rate” er hvor hurtigt udviklerne “brænder en estimeret time af”

BURN DOWN CHART



Eksempel

4 dage på 12 moduler / 2 personer - tasks på 24 timer



ANVENDELSE AF BURN DOWN CHART

- Hvor mange timer koster projektet?
 - EKSEMPEL:
Et projektet er vurderet til at koste: 120 timer, har 3 mand, der hver arbejder 4 timer pr dag, kan ideelt kunne færdiggøres på: $120 \text{ timer} / (3 \text{ mand} * 4 \text{ timer}) = 10 \text{ dage}$
Dvs. på **y-aksen 120 timer** og på **x-aksen 10 dage**.
- Burn Down Chart - skal opdateres hver dag!
 - For at være brugbart beslutningsgrundlag bør "burn down chart" opdateres dagligt
- Udnævn en projektleder
 - Opdateringen af "burn down chart", kontakt med til vejleder, satusmøder og mange andre ting kan nemt blive glemt. Det er derfor en super god ide at udnævne en projektleder som har hovedansvaret for alle disse projekt

Vejledere og “kunder”

I digitalt design har i muligvis ikke kunder. Jeres vejleder overtager også denne rolle...