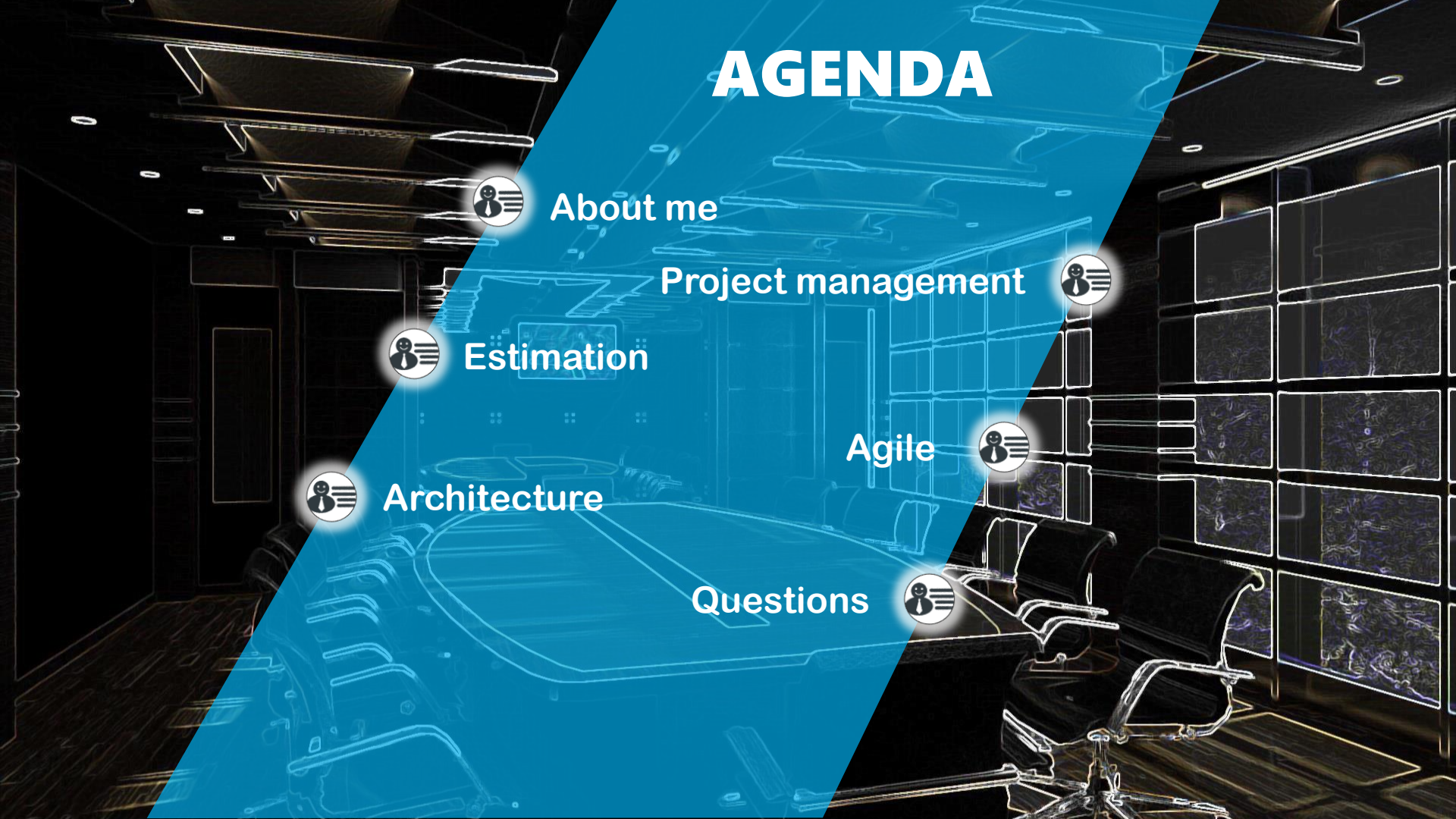
frAGILE IS JUST A WORD

Reflections about estimation, architecture, and agile issues

TDT4290 Customer Driven Project

October 16, 2017

http://www.houseofbusiness.com/wp-content/uploads/2015/03/iStock_000057564776_Large.jpg


AGENDA
About me
Project management
Estimation
Agile
Architecture
Questions

About me

2009

2009 -

BEKK

Lånekassen

2012 -

@ <3

Safurudin Mahic

“

The project work will be evaluated based on the quality of the project report and presentation delivered at the end of the course and the students' reflections on the project work.

How the group actually has worked, technical problems, customer behaviour and availability, etc. are a part of the reflections report.

The group is asked to deliver a 3-4 page report, reflecting upon their experiences during the project and reflecting upon what they had learned and how they could have done things differently. The *focus of this part is on the process rather than the product*.

Introduction

The following criteria are evaluated in an integrated way:

- Whether the group has solved the given assignment, according to the customer's objectives of the project.
- Team work efficiency and the team dynamics
- Team work process improvement efforts
- Reasonable grounds for decisions taken.
- Visibility of limitations done.
- The students' ability to reflect on the process during the project.
- Layout and structure readability.
- Logical flow in the report.

Making software is (unfortunately) not just about writing code.

It is equally much as to make informed decisions. This is why there are developed methods both in risk management, estimation, project management, architecture management.

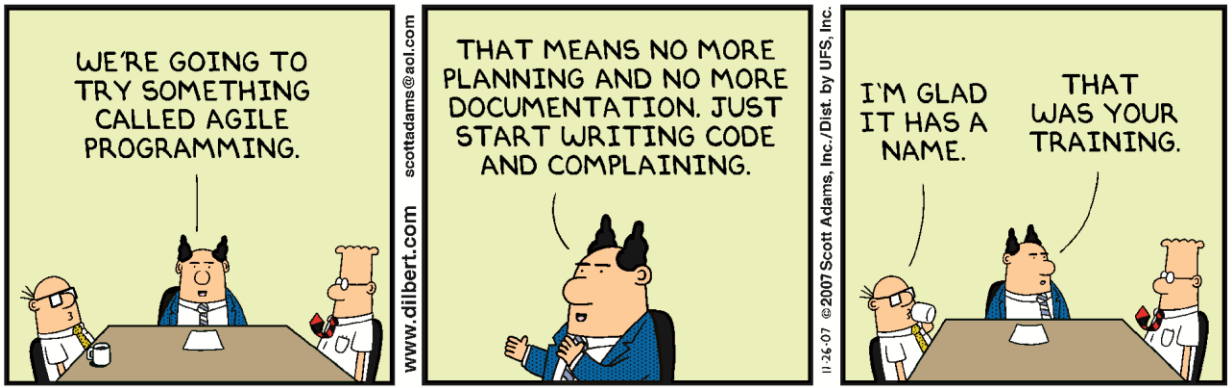
Knowing about various principles, methodologies, patterns and good practices helps.

Think about how you can document and learn from your process.

Be aware of the choices you make – know why you make them

Communication is the most important factor

Project management


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

“

Project refers to efforts made to achieve a **defined goal**, usually within a planned **time** and resource framework. The term is used for activities ranging from simple school projects to eg. engineering of complex oil platforms in the North Sea. As a tool for keeping projects within the framework given while achieving the goal, **theories** and **methods** have been developed for how to implement them in the best possible way.

Project management is about how to apply good practices in the fields of

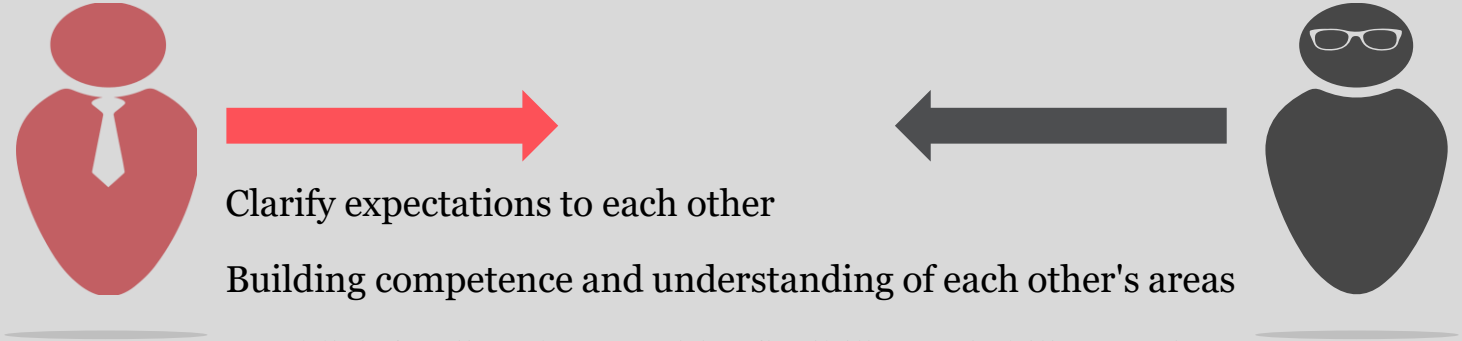
- Process
- Estimation
- Excecution
- Architecture
- Many others

to achieve a goal

> “Project management is “the application of knowledge, skills, tools, and techniques to project activities in order to meet project requirements” (PMI*, Project Management Body of Knowledge (PMBOK® Guide))
> *The Project Management Institute (PMI) is an international professional society.

BUSINESS / CUSTOMER & IT MUST COOPERATE

Business **IT**



Clarify expectations to each other

Building competence and understanding of each other's areas

Establish funding that provides flexibility and ability to adapt

Provide direction and focus

Participate in discussions and make informed decisions - even when effects and execution requirements are uncertain

See opportunities and formulate needs

Visualise opportunities

Turn business needs into good solutions

There are no silver bullets

Every single project is

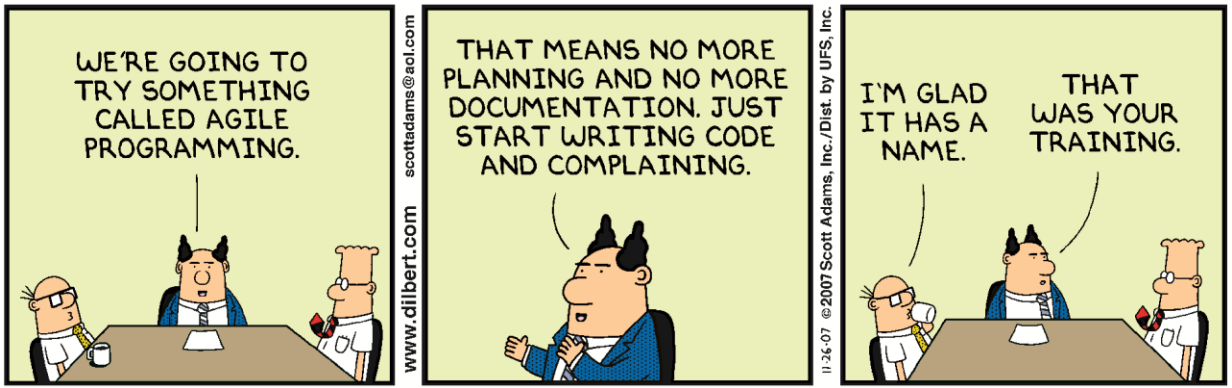
agile

ˈadʒʌɪl/

adjective

able to move quickly and easily.
"Ruth was as agile as a monkey"

relating to or denoting a method of project management, used especially for software development, that is characterized by the division of tasks into short phases of work and frequent reassessment and adaptation of plans.
"agile methods replace high-level design with frequent redesign"


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

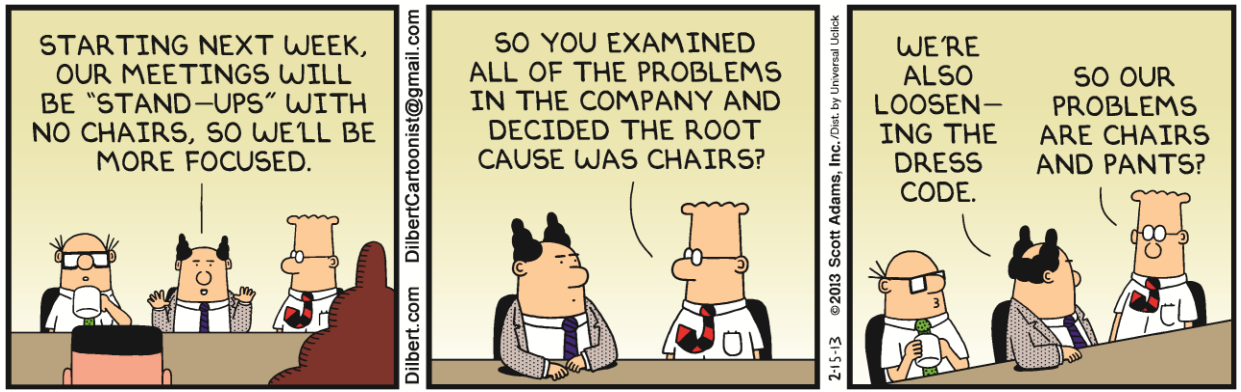
Manifesto for Agile Software Development

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

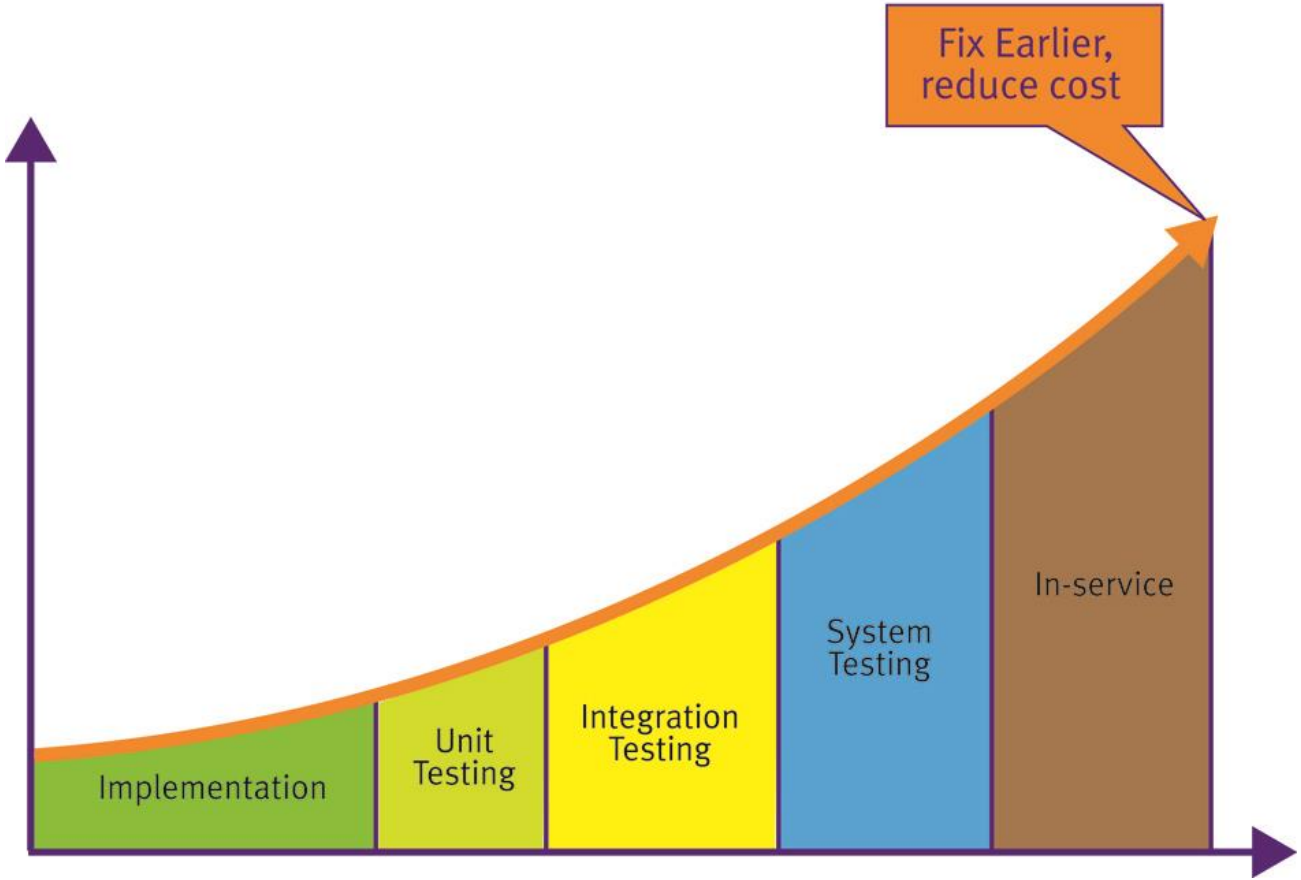
That is, while there is value in the items on
the right, we value the items on the left more.

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


STARTING NEXT WEEK, OUR MEETINGS WILL BE "STAND-UPS" WITH NO CHAIRS, SO WE'LL BE MORE FOCUSED.
SO YOU EXAMINED ALL OF THE PROBLEMS IN THE COMPANY AND DECIDED THE ROOT CAUSE WAS CHAIRS?
WE'RE ALSO LOOSEN-ING THE DRESS CODE.
SO OUR PROBLEMS ARE CHAIRS AND PANTS?
Dilbert.com DilbertCartoonist@gmail.com
© 2013 Scott Adams, Inc./Dist. by Universal Uclick
2-15-13


Extreme Programming
Scrum
Lean
Crystal
DSDM
++

The Cost of Defects



http://www.embeddedinsights.com/channels/2012/03/19/unit-test-tools-and-automatic-test-generation/


© Instagram/thenotoriousmma

Tip #1

Make sure you really understand the goal and the vision for the proje

Lånekassen | Dine sider

English | Nynorsk | A A

Meny | Hei, Hans! | Din profil | Logg ut

Forside

Postkasse
Du har 1 ulest(e) brev eller e-post

Dine søknader

Din gjeld
Din samlede gjeld er 256 332 kr

Din regning
Neste forfall er den 15. september beløpet er på 1 255 kr

Søknader og skjemaer

- Stipend og lån
- Tilbakebetaling
- Frister

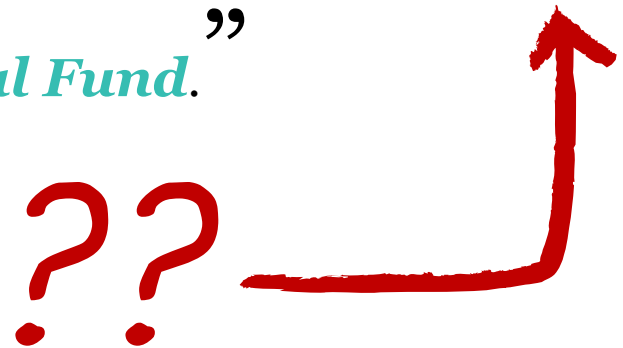
Snarveier

- Nettstedskart
- Personvern og cookies

VI GJØR UTDANNING MULIG

Goal and vision for project

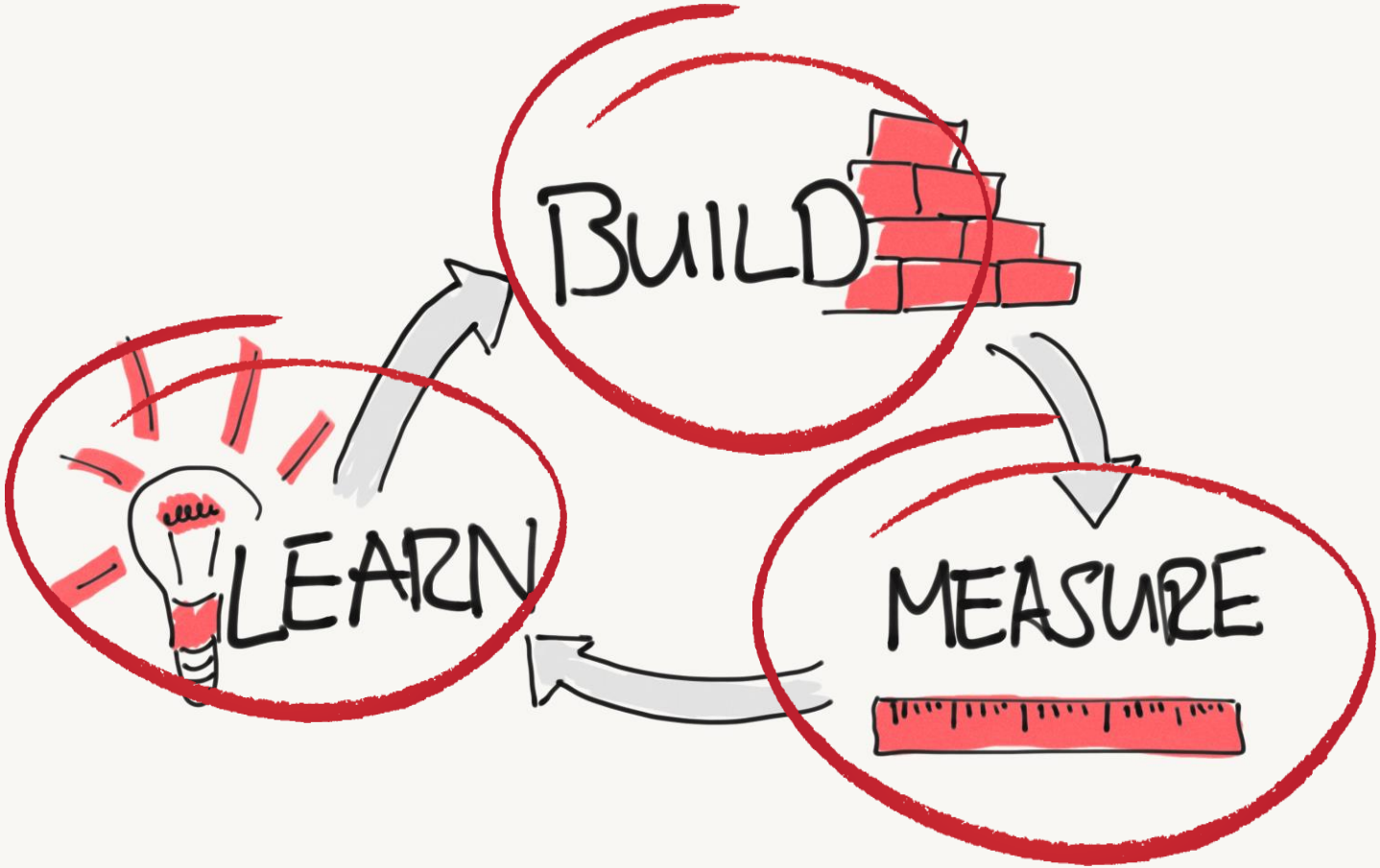
“The portal is supposted to be ***self explanatory***, so that the users are ***self-serviced*** and ***don’t need to contact Norwegian State Educational Fund***.”



“The best way to make sure your product meets the needs of your target audience is to expose your designs to the scrutiny of your users.”

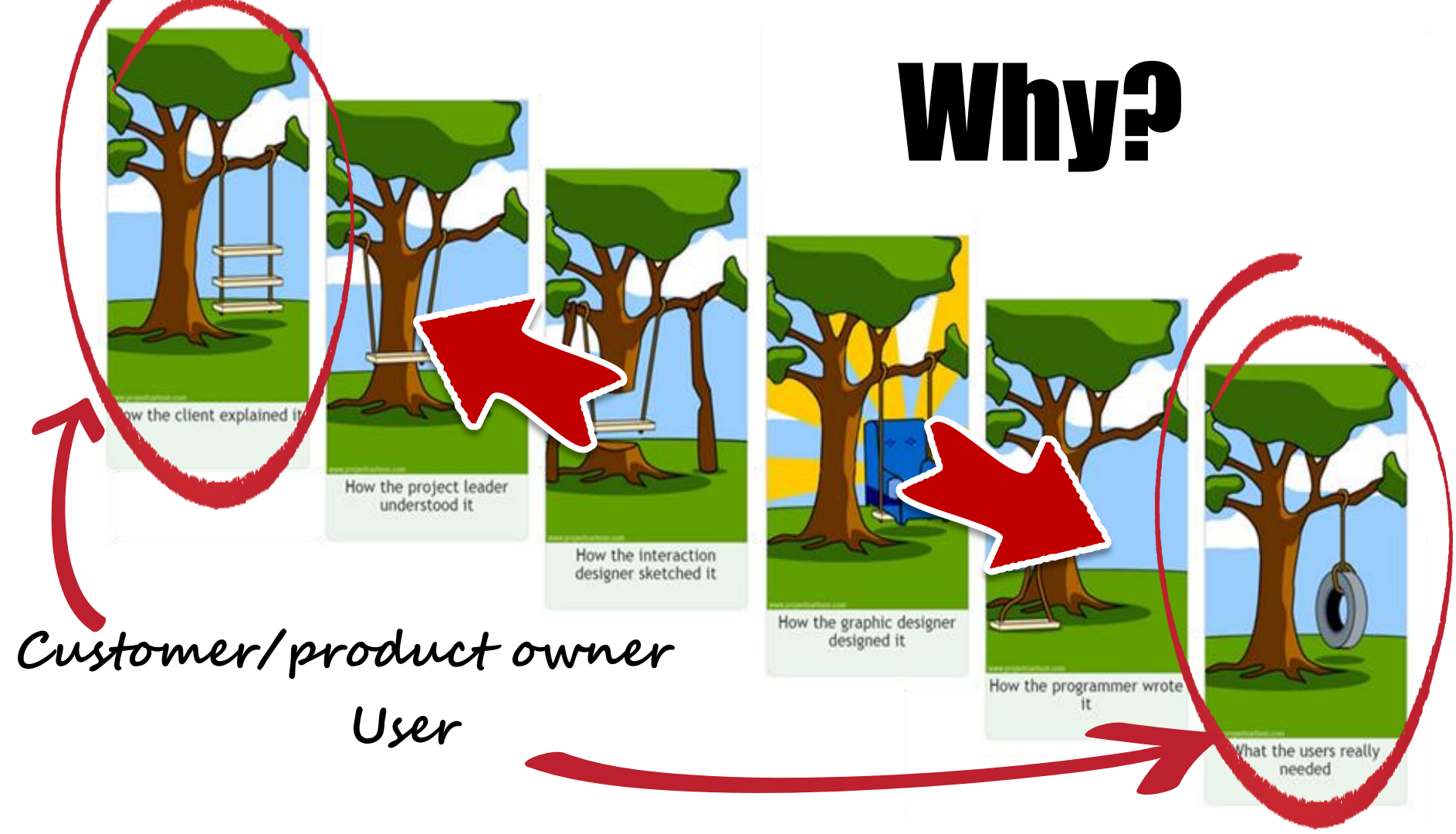
Apple Human Interface Guidelines

How?



https://vimeo.com/114878579

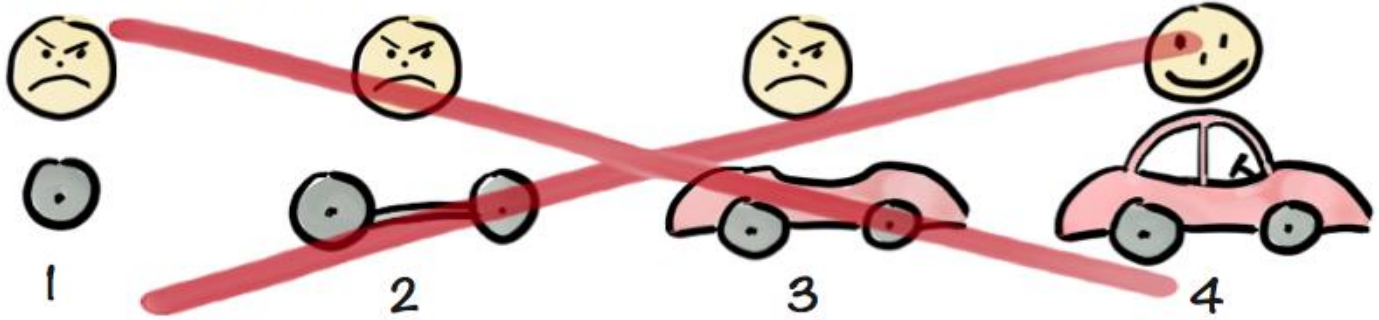
Why?



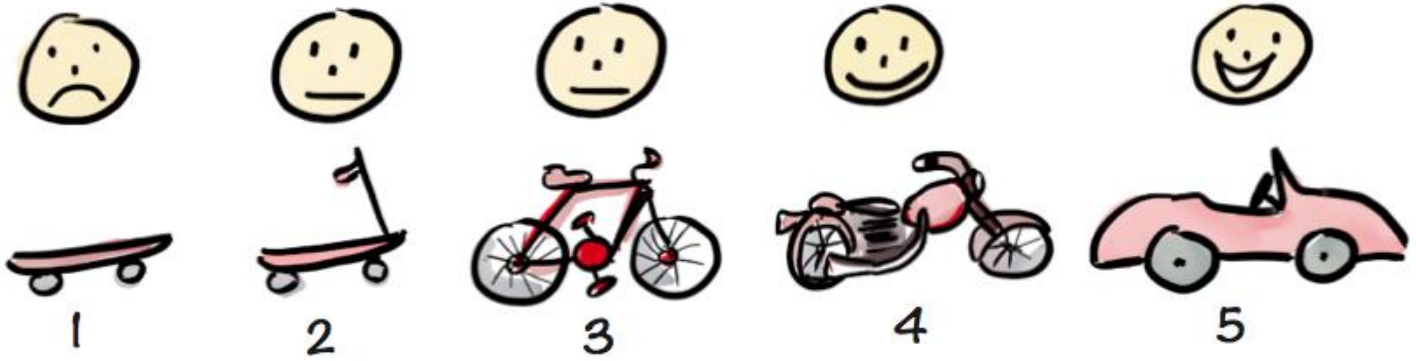
Customer/product owner

User

Not like this....



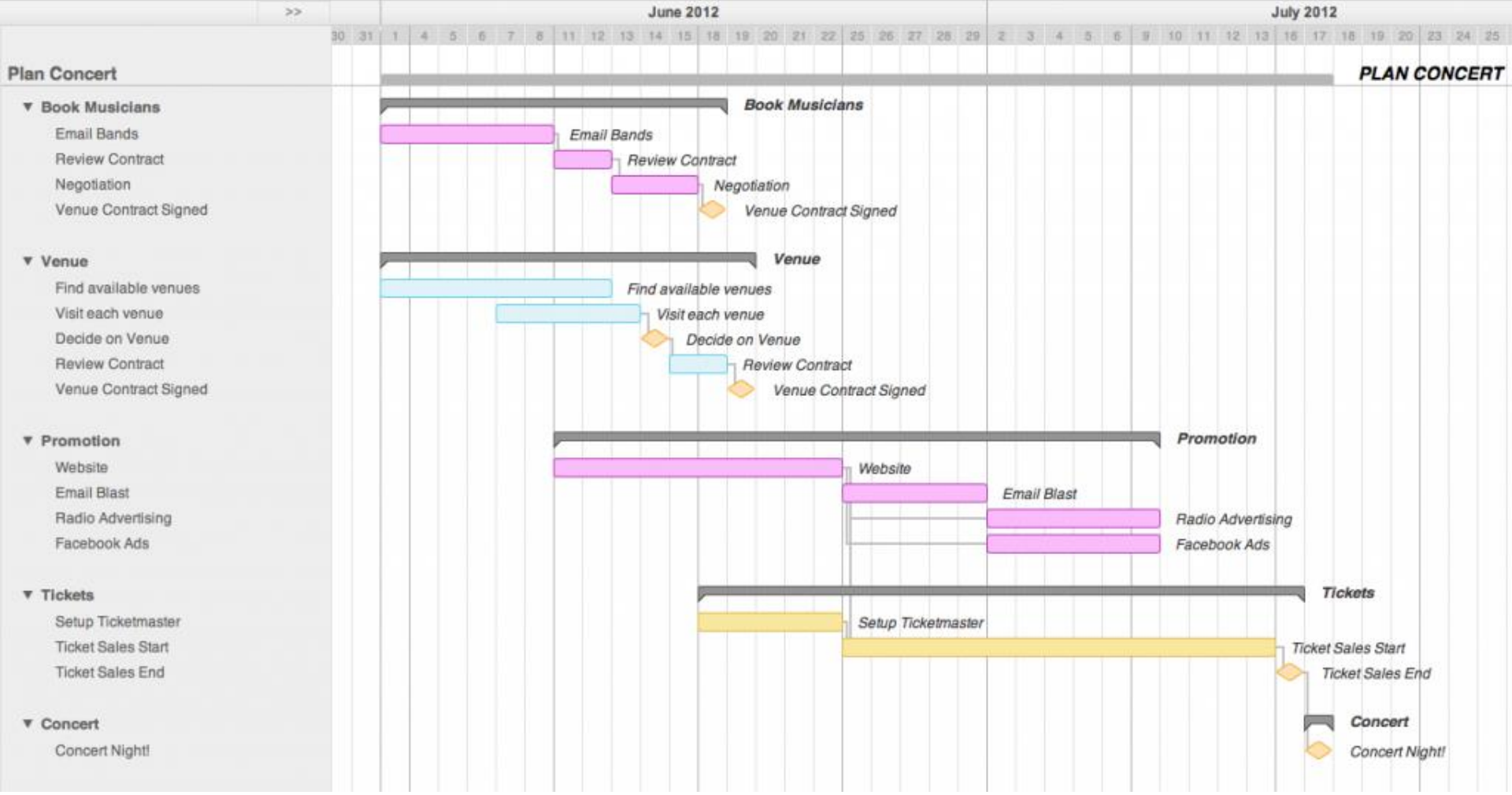
Like this!



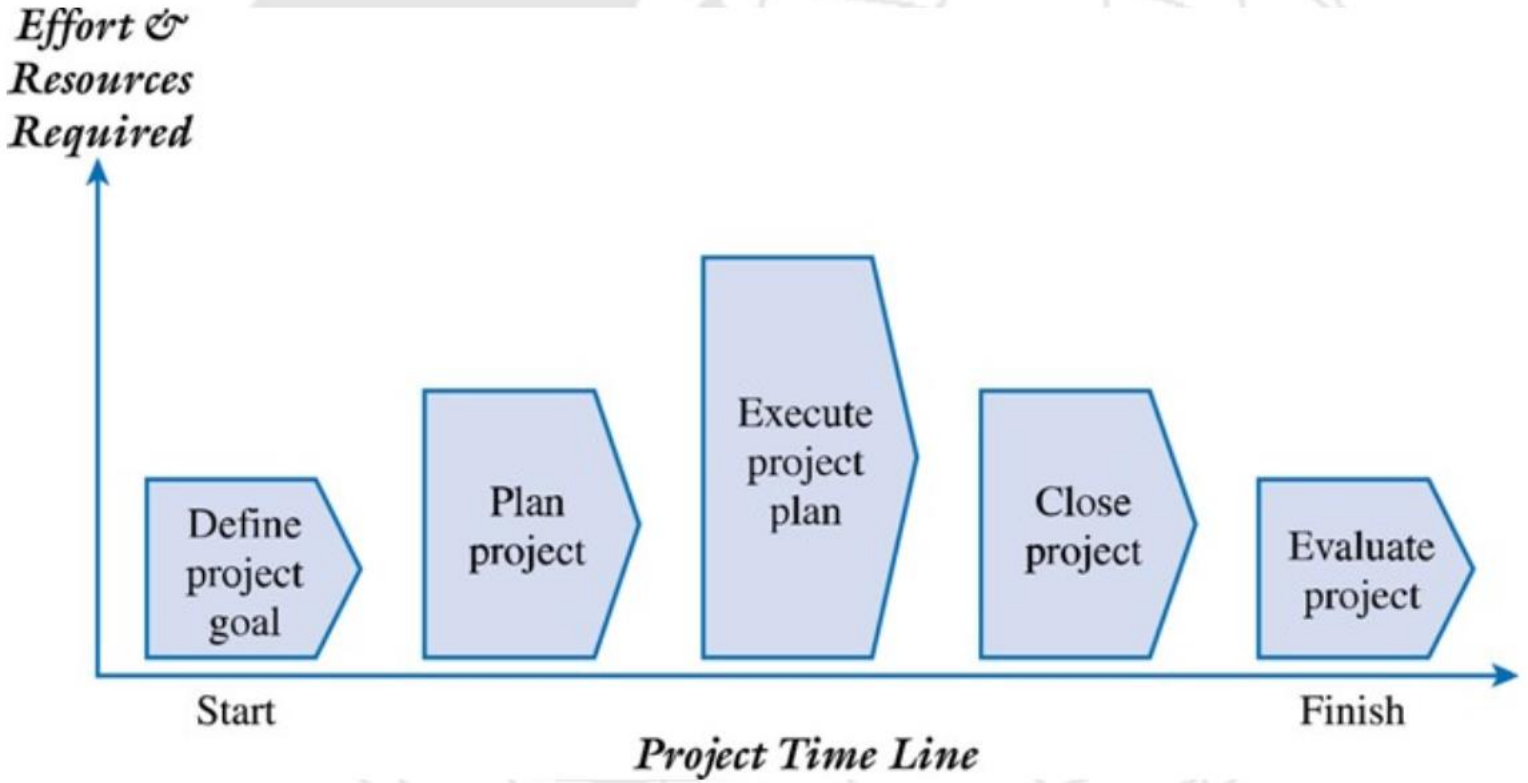
Henrik Kniberg

Tip #2

Create an revise plan for the project


>>
June 2012
July 2012
30 31 1 4 5 6 7 8 11 12 13 14 15 18 19 20 21 22 25 26 27 28 29 2 3 4 5 6 9 10 11 12 13 16 17 18 19 20 23 24 25
Plan Concert
PLAN CONCERT
Book Musicians
Email Bands
Review Contract
Negotiation
Venue Contract Signed
Venue
Find available venues
Visit each venue
Decide on Venue
Review Contract
Venue Contract Signed
Promotion
Website
Email Blast
Radio Advertising
Facebook Ads
Tickets
Setup Ticketmaster
Ticket Sales Start
Ticket Sales End
Concert
Concert Night!

Generic Project Life Cycle



Prosjektveiviseren

https://www.prosjektveiviseren.no/

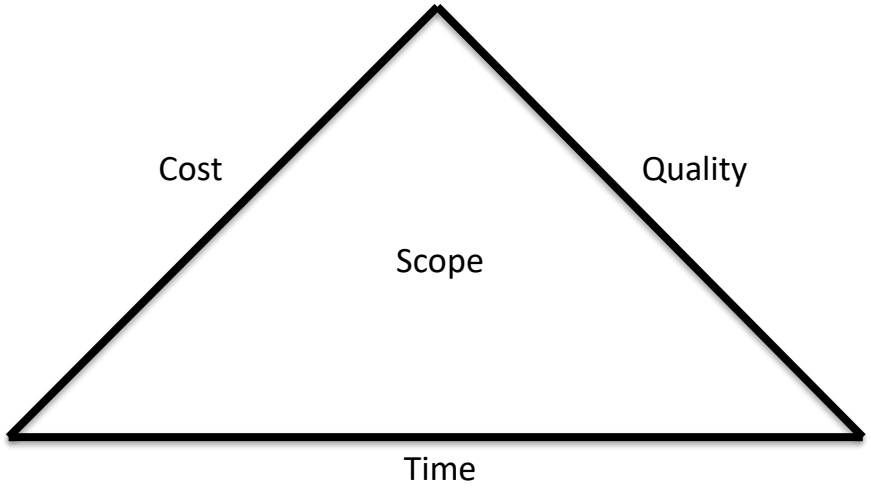
Estimation

Two questions for you

- When do we estimate?
- Why do we estimate?


I NEED A BUDGET ESTIMATE FOR MY PROJECT, BUT I DON'T HAVE A SCOPE OR A DESIGN FOR IT YET.
Dilbert.com DilbertCartoonist@gmail.com
OKAY, MY ESTIMATE IS $3,583,729.
12-7-09 ©2009 Scott Adams, Inc./Dist. by UFS, Inc.
YOU DON'T KNOW ANYTHING ABOUT MY PROJECT.
THAT MAKES TWO OF US.

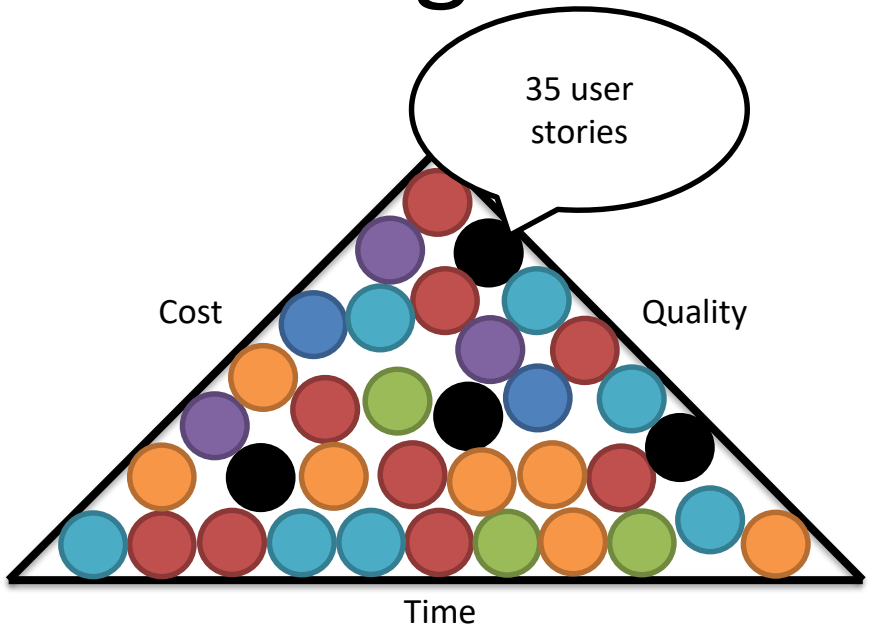
Project management triangle



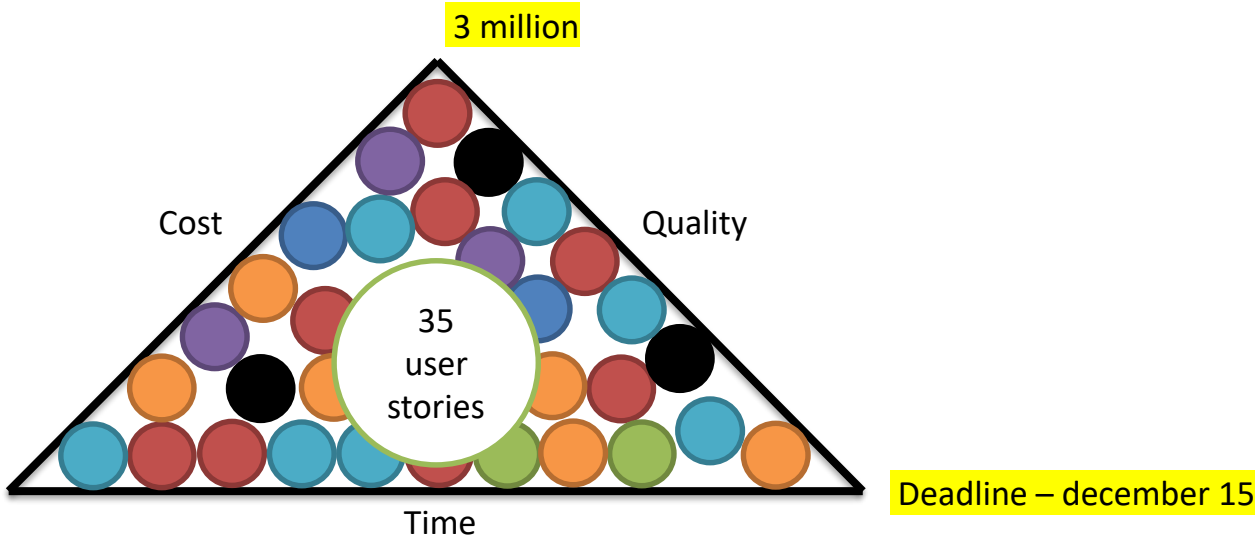
https://en.wikipedia.org/wiki/Project_management_triangle

The triangle refers to the “good – cheap – fast” dilemma. Too often, customers ask for a job of excellent quality, at an economical price, at a pace that requires warp speed. So the adage goes like this: “You can have two out of three - good, fast or cheap – but you can’t have all three.”

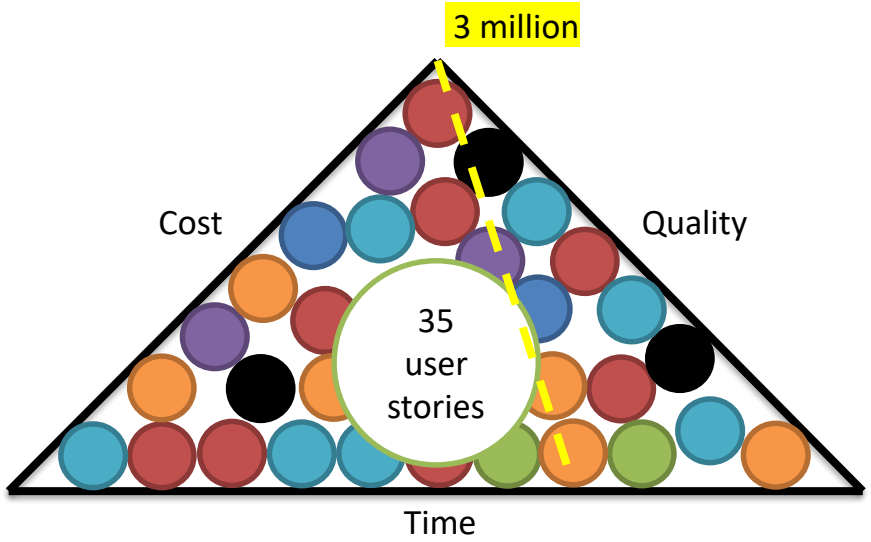
Project management triangle



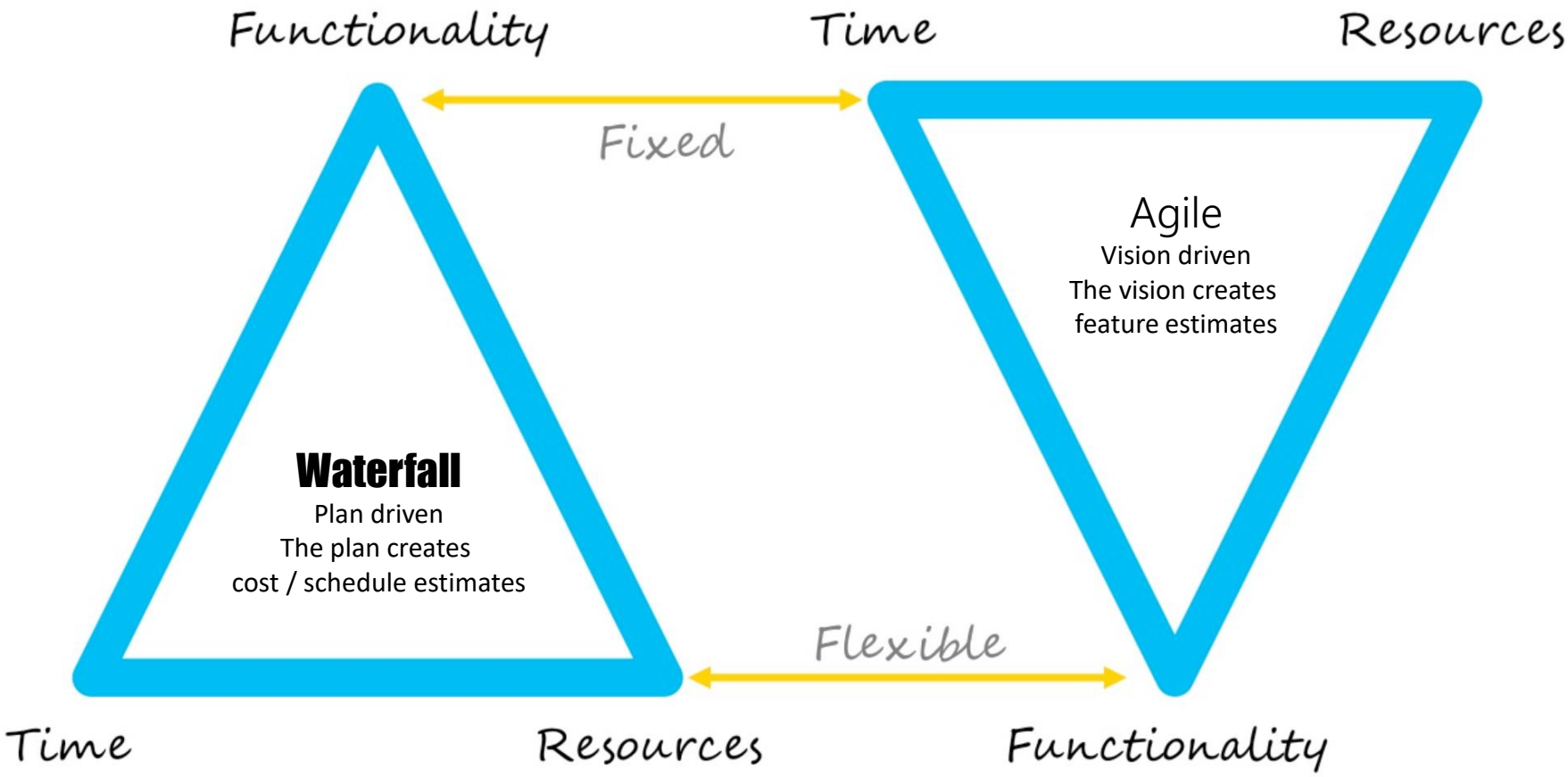
Project management triangle

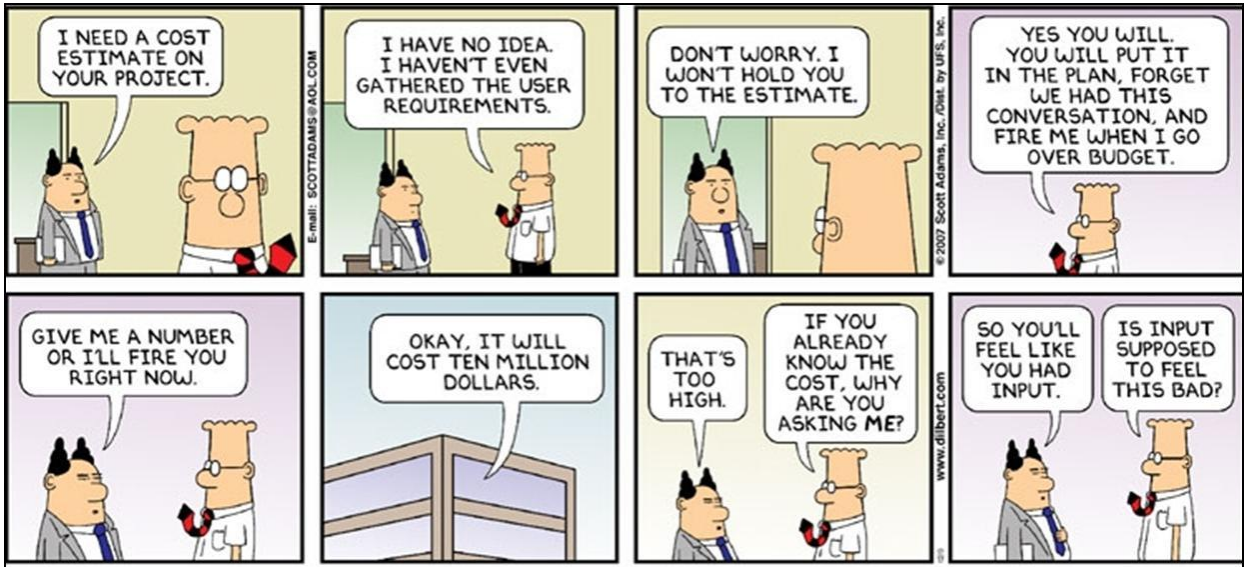


Project management triangle



Deadline – november 23


Functionality
Time
Resources
Fixed
Agile
Vision driven
The vision creates
feature estimates
Waterfall
Plan driven
The plan creates
cost / schedule estimates
Flexible
Time
Resources
Functionality



“We know what we want. Can you estimate how long it will take to build?”

“We need to get these requirements nailed down before we can start development.”


NAT·
MDCCC
XXXIII
OB·
MDCCC
XCVI
ALFR·
NOBEL

+


THE GREATEST NATION ON THE FACE OF THE EARTH
THE UNITED STATES OF AMERICA
THIS NOTE IS NOT LEGAL TENDER FOR ANY DEBTS PUBLIC OR PRIVATE
1000000MUSA
WASHINGTON, D.C.
1000000MUSA
AAC © Copyright
MISS LIBERTY
ONE MILLION DOLLARS
SERIES LIBERTY
First Vice President of the United States

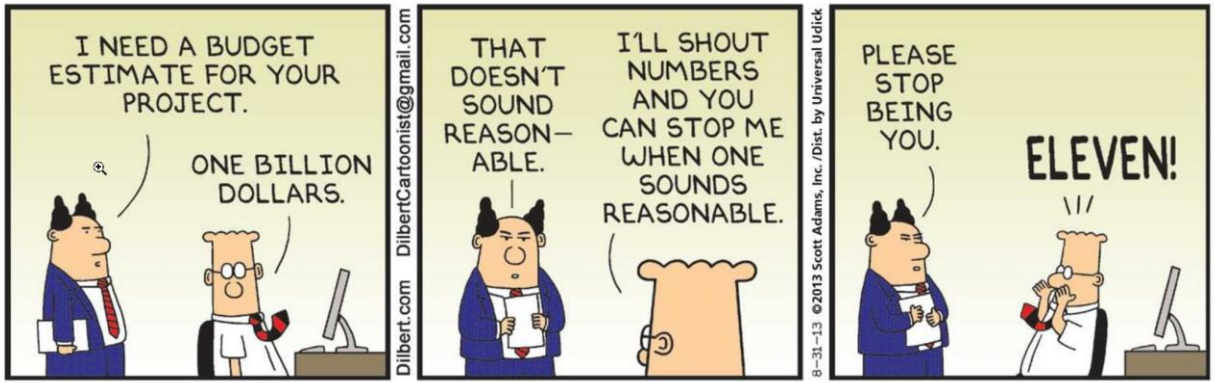
Cost

Benefit

Beneficial return

Tip #3

Create an revise a scope and estimate for the project


I NEED A BUDGET ESTIMATE FOR YOUR PROJECT.
ONE BILLION DOLLARS.
THAT DOESN'T SOUND REASON-ABLE.
I'LL SHOUT NUMBERS AND YOU CAN STOP ME WHEN ONE SOUNDS REASONABLE.
PLEASE STOP BEING YOU.
ELEVEN!
Dilbert.com DilbertCartoonist@gmail.com
8-31-13 ©2013 Scott Adams, Inc. /Dist. by Universal Uclick

Why do businesses need cost estimates?

- To screen investment opportunities
- Plan expenditure cycles
- Obtain Financial sanction both internally and from any joint venture partners
- Budget , Control and Reporting on projects progress
    - Internally
    - Fiscal reporting

An estimate is a guess and you need to convince your audience that it is a good guess

Importance of Good Estimates
Correct Project Cost
Cost OverEstimate
• Project Cancelled- Lost economic opportunity
• Project proceeds – excess capital tied up

Underestimate
• Project proceeds
o Capital overspend
o Project economic criteria not achieved
o Loss of confidence in company
o Share/Stock price may be effected
o Adversely effect financing of future investment opportunities

Estimate Key Components

An estimate should comprise of:

- A well worked base estimate
- A realistic contingency budget
- A credible accuracy range that reflects the project uncertainties and risks (threats and opportunities)

Definitions – Contingency / Management Reserve

Contingency is an integral part of the estimate: "An amount added to an estimate to allow for items, conditions, or events for which the state, occurrence, or effect is uncertain and that experience shows will likely result, in aggregate, in additional costs." (Association for the Advancement of Cost Engineering , AACE)

"The amount of funds, budget, or time needed above the estimate to reduce the risk of overruns of project objectives to a level acceptable to the organization." (Project Management Body of Knowledge, PMBOK)

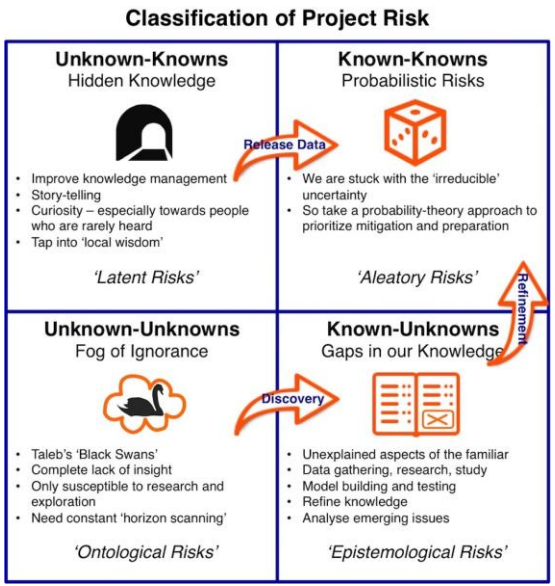
Cost Engineering Terminology", Recommended Practice 10S-90, AACE International, WV, rev. 2007

Establishing the Estimate

Technical Basis
- Process design
- Codes & standards
- quantities

Execution Plan
- contracting strategy
- Schedule
- Resources

Estimate Basis
- Unit rates
- Productivity assumption
- Owners management
- Exclusions



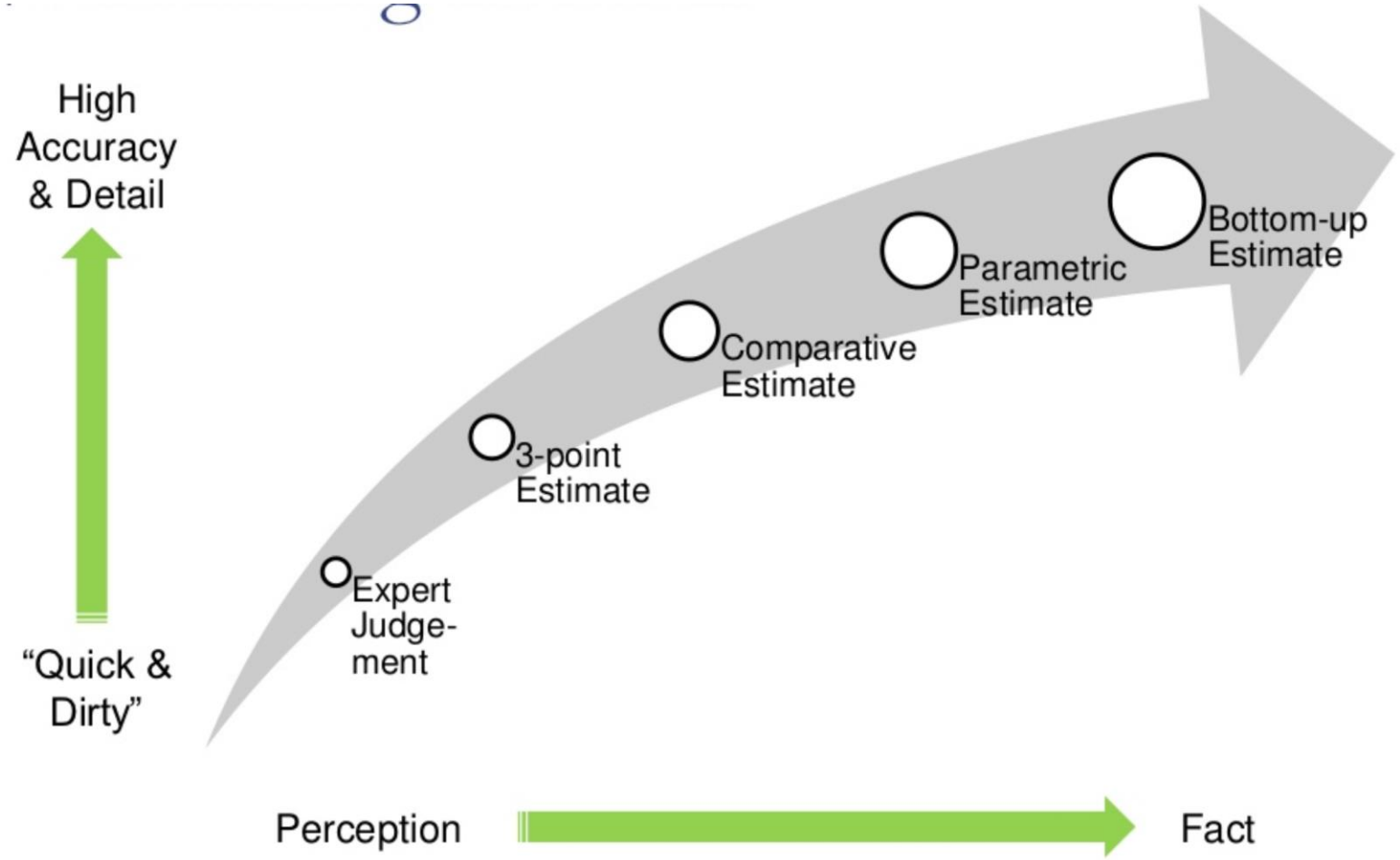
Issues, Risks, and Challenges:

1. **Known Knowns**: Items with which we have experience or knowledge and for which we are comfortable with how to address and manage them.
2. **Known Unknowns**: Items that we know about but do not have enough information, experience, or knowledge to address.
3. **Unknown Unknowns**: Items that we are completely ignorant of, are unexpected when they arise, and are typically adverse to a project's outcome.

A typical classification of risks is based on the level of knowledge about a risk event's **occurrence**(either known or unknown) and the level of knowledge about its **impact** (either known or unknown). This leads to four possibilities (Cleden, 2009):

- Known–knowns (knowledge)
- Unknown–knowns (impact is unknown but existence is known, i.e., untapped knowledge)
- Known–unknowns (risks), and
- Unknown–unknowns (unfathomable uncertainty).

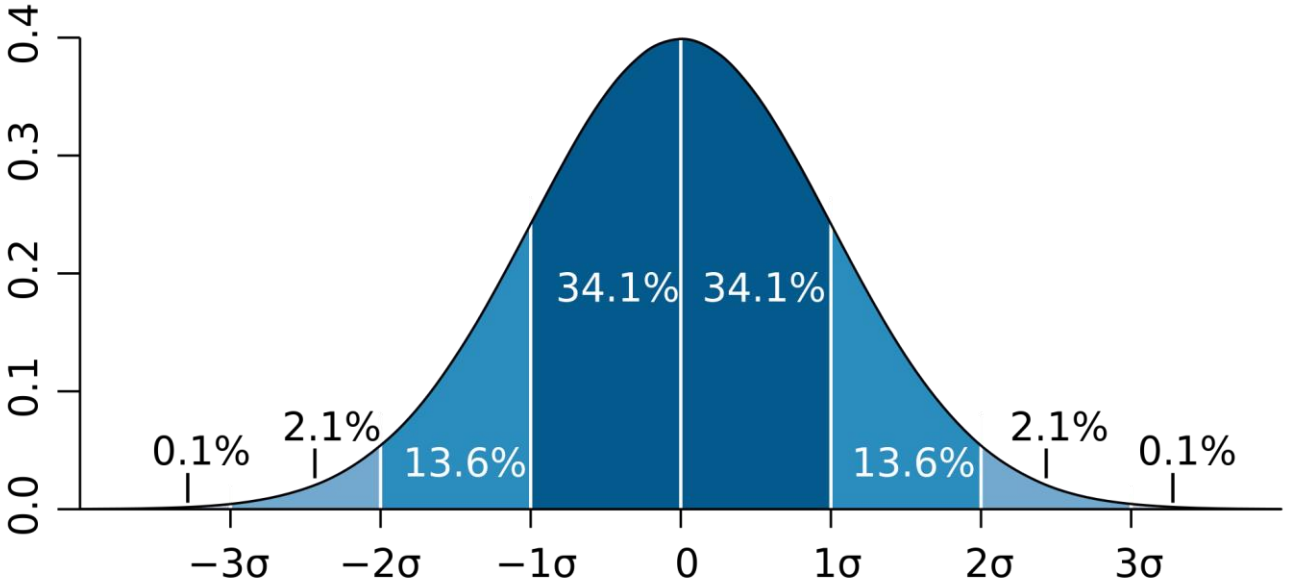
Cleden, D. (2009). *Managing project uncertainty.* Farnham, UK: Gower.
Hillson, D. (2005). Why Risks Turn into Surprises, *Risk Doctor Briefings* [Electronic Version] No.16. Retrieved from http://www.risk-doctor.com/pdf-briefings/risk-doctor16e.pdf



https://www.slideshare.net/ianjseath/project-estimating-15032124

Normal Distribution Shape is determined by its mean (μ) and standard deviation (σ) Probability is associated with area under the curve. Since the distribution is symmetrical, the following probability rules of thumb apply

- About 68 percent of all the values will fall between +1 σ of the mean
- About 95 percent of all the values will fall between +2 σ of the mean
- About 99 percent of all the values will fall between +3 σ of the mean



http://www.ap-learning.net/dmdocuments/2008-03-01-Pharmaceutical%20Engineering-Stage%20Gates.pdf

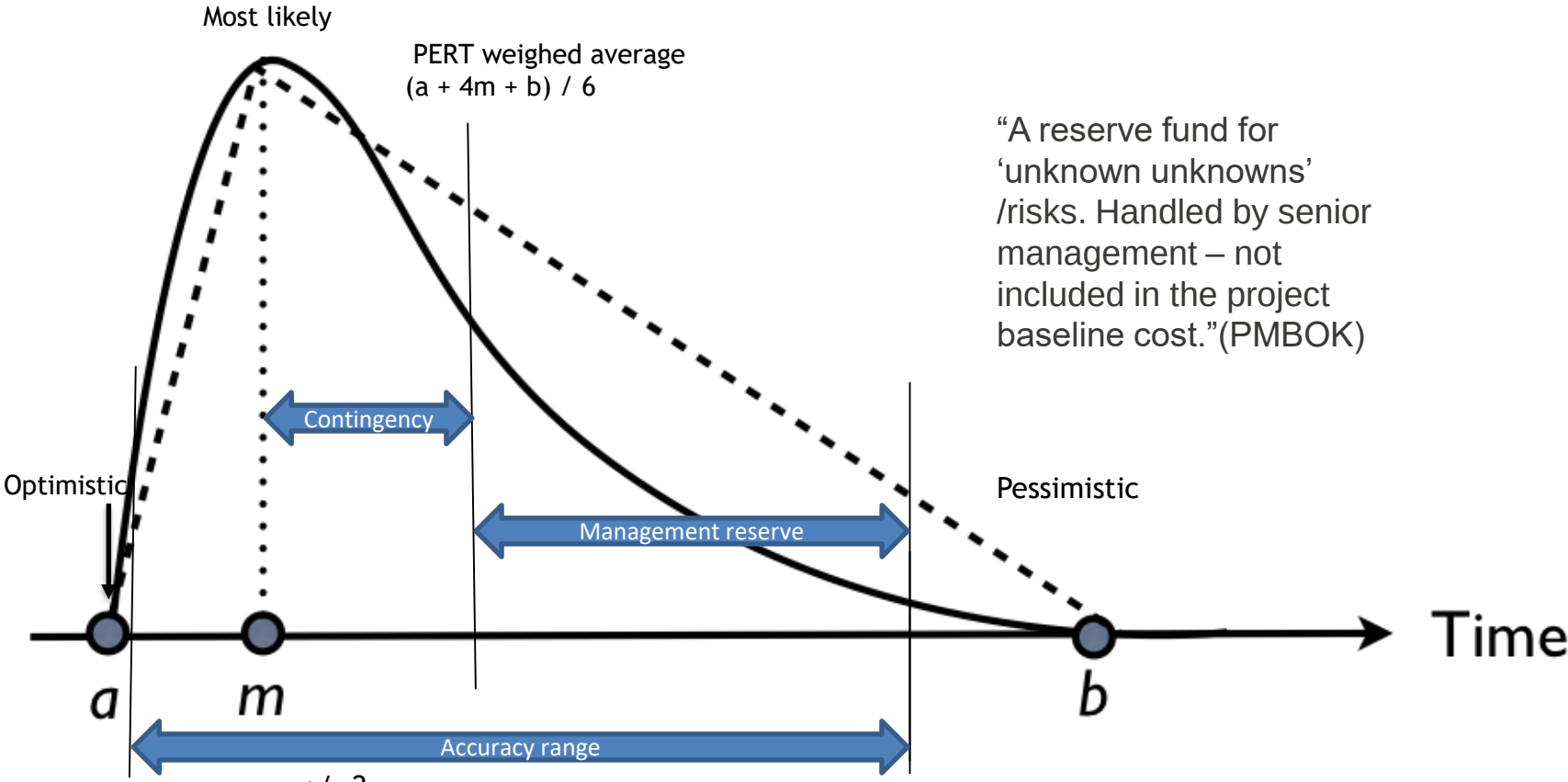
PERT distribution uses a three-point estimate where

- a denotes an optimistic estimate
- m denotes a most likely estimate
- b denotes a pessimistic estimate

PERT Mean = (a + 4m + b) / 6
PERT Standard Deviation = (b - a) / 6

Analytic Method for Probabilistic Cost and Schedule Risk Analysis Final Report,
https://www.nasa.gov/pdf/741989main_Analytic%20Method%20for%20Risk%20Analysis%20-%20Final%20Report.pdf

Estimation techniques - PERT and Critical Path Method



"A reserve fund for 'unknown unknowns' /risks. Handled by senior management – not included in the project baseline cost."(PMBOK)

http://www.pmknowledgecenter.com/node/59

Task	Predecessors	a	m	b
A	none	2	5	8
B	A	1	2	9
C	A	0,25	0,5	3,75
D	B	1	1	7
E	B and C	1	2	9
F	D and E	1	3	11
Total		**6,25**	**13,5**	**47,75**

σ	**6,92**
PERT weighted average	**17,00**
95% confidence interval	**30,83**

https://www.slideshare.net/SerdarTemi/10-project-riskmanagement

Value and Waste

In assessing a process, it is important to understand what activities in the process actually add value to the end result. All other activities are wasteful.

CVA (Customer Value Added – or just VA for Value Added): adding form fit or function to a product or service, an activity that the customer would be willing to pay for in isolation if they knew it was being done – e.g. Creating code, implementing functionality.

BVA (Business Value Added – non-negotiable waste): an activity that is required to operate the business but the customer is unwilling to pay for – e.g. Budget tracking, code documentation.

NVA (Non-Value Added): an activity that is not required by the business nor is the customer willing to pay for – e.g. Waiting for resource allocation, requirements documents, estimation

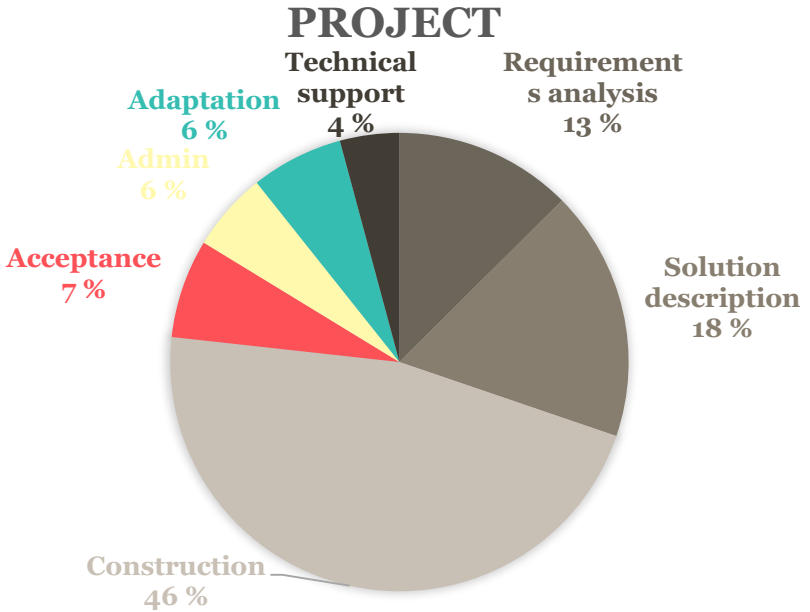
Principles of Agile Estimation

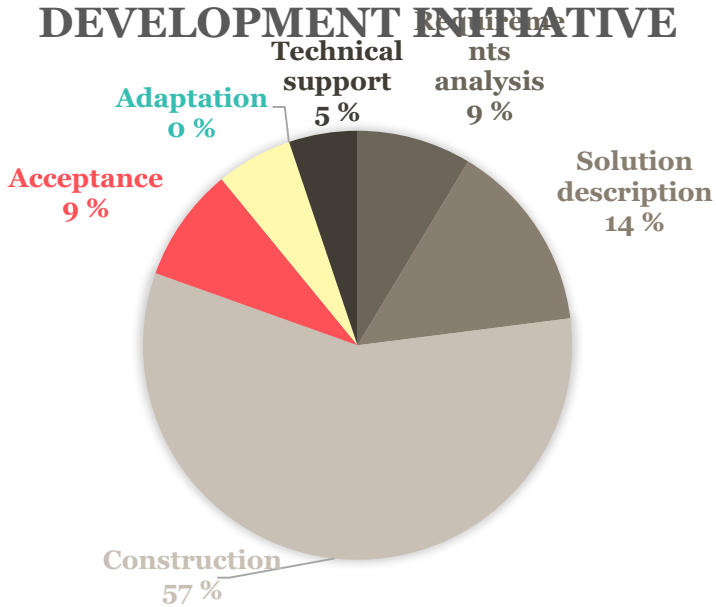
Agile estimation techniques are collaborative. All appropriate people are included in the process. For example the whole Scrum team participates in estimating effort of Product Backlog Items. Collaborative techniques are also designed so that it is impossible to blame someone for an incorrect estimate: there is no way to trace who estimated what.

Agile estimation techniques are designed to be fast (-er than traditional techniques) and deliberately trade off accuracy. We are not trying to learn to predict the future or get better at estimation. Instead, we recognize that estimation is a non-value-added activity and minimize it as much as possible.

Most Agile estimation techniques use relative units. This means that we don't try to estimate dollars or days directly. Instead, we use "points" or even qualitative labels and simply compare the items we are estimating to each other. This takes advantage of the human capacity to compare things to each other and avoids our difficulty in comparing something to an abstract concept (such as dollars or days).

http://www.agileadvice.com/2015/10/13/agilemanagement/9-agile-estimation-techniques/





Discipline	Project	Development initiative
Requirements analysis	27 %	15 %
Solution description	38 %	25 %
Construction	100 %	100 %
Acceptance	15 %	15 %
Admin	12 %	10 %
Adaptation	14 %	0 %
Technical support	9 %	9 %
Total	215 %	174 %

Dicipline	Description
Requirements analysis	Analysis and description of the functional scope of the project. Detailing functionality Product Owner time
Solution description	Technical description of the solution (2 steps) Step 1 – detailed enough to give 3-point estimates Step 2 – detailed enough for development
Construction	System development + unit tests Stabilized functionality by dedicated testers
Acceptance	Test management Plan system and acceptance tests Execute system and acceptance tests Document routines and procedures for operations
Adaptation	Internal training and communication Product verification
Technical support	Environment setup Deploy setup Test of deployment plan Execution of deployment
Admin	Project management Team management

Øker studiestøtten til elleve måneder

Regjeringen foreslår å utvide studiestøtten fra neste sommer. Det blir en ekstra måned med studielån, og totalt øker studiestøtten fra 94.000 kroner i år til 105.600 kroner. – Vi er veldig glade, sier NSO.

VELDIG FORNØYD: – Dette er et historisk gjennomslag for studentstøtten, sier leder i Norsk Studentorganisasjon Ola Magnussen Rydje.

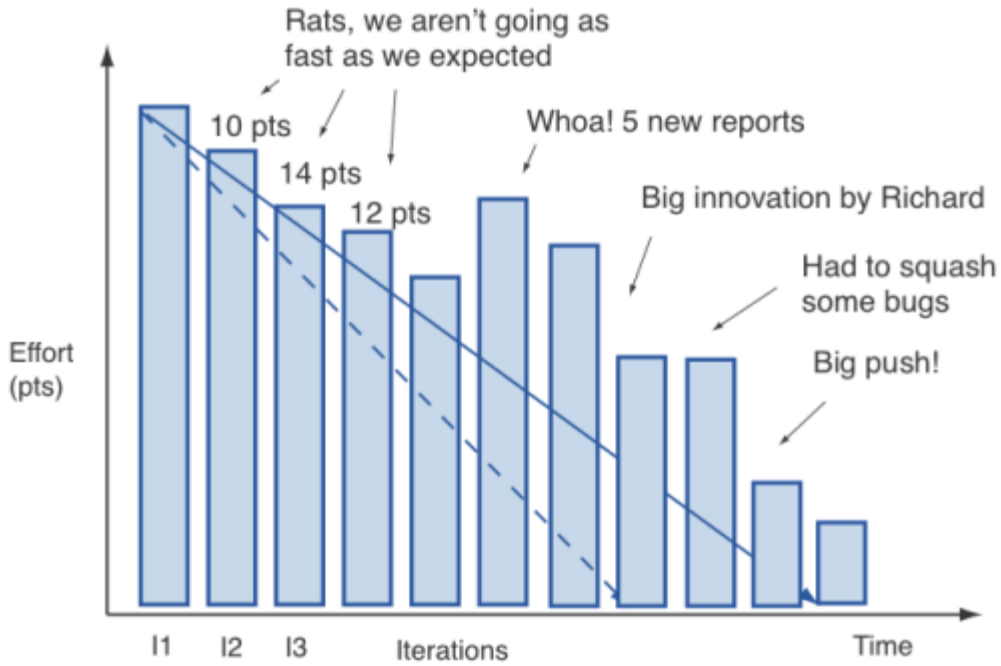
Anne Lise Stranden

(DinePenger) Publisert: 11:34 - 14.10.2013, Oppdatert: 11:51 - 14.10.2013

En ekstra måned. Det legges opp til elleve måneder utdanningsstøtte per undervisningsår til studenter og andre som får støtte etter samme regelverk. Første ekstra måned med støtte vil bli juni 2015.

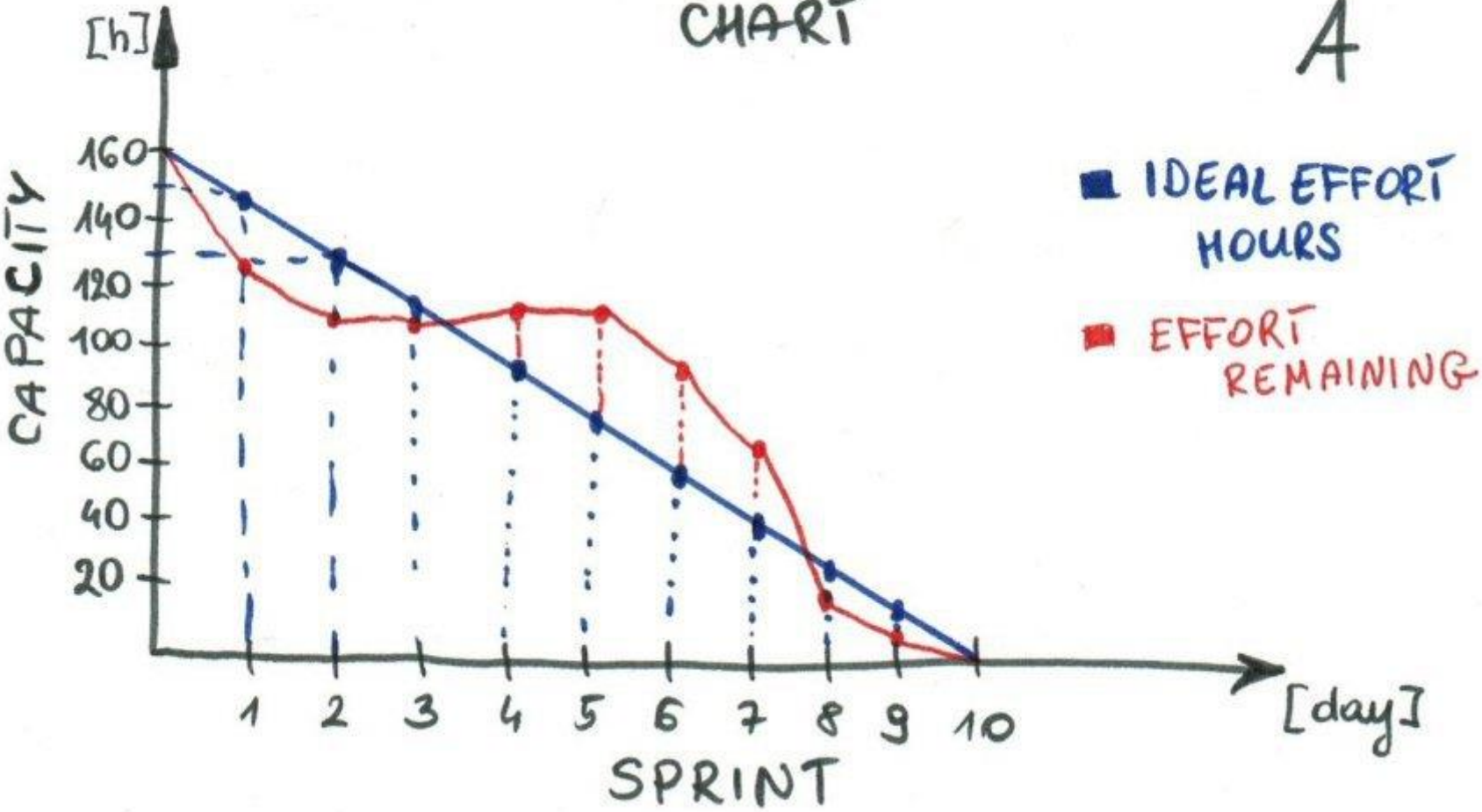
Resource budget

Person	34	35	36	37	38	39	40	41	42	43	44	45	46	47	#
Roger	25	25	25	25	25	25	25	25	25	25	25	25	25	25	325
Thomas	25	25	25	25	25	0	25	25	25	25	25	25	25	25	300
Cathrine	25	25	25	25	25	25	25	25	25	25	25	25	25	25	325
Steven	25	25	25	25	25	25	25	25	25	25	25	25	25	25	325
Anne	25	25	25	25	25	25	25	25	25	25	25	25	25	25	325
Budgeted	125	125	125	125	125	100	125	125	125	125	125	125	125	125	1 600
Actual	110	125	125	125	125	100	125	100							
Diff	-15	0	0	0	0	0	0	-25							-40


Rats, we aren't going as fast as we expected
10 pts
14 pts
12 pts
Whoa! 5 new reports
Big innovation by Richard
Had to squash some bugs
Big push!
Effort (pts)
I1
I2
I3
Iterations
Time

SPRINT BURN-DOWN CHART

A



Tip #4

Track progress / speed / hit rate

Agile

AGILE

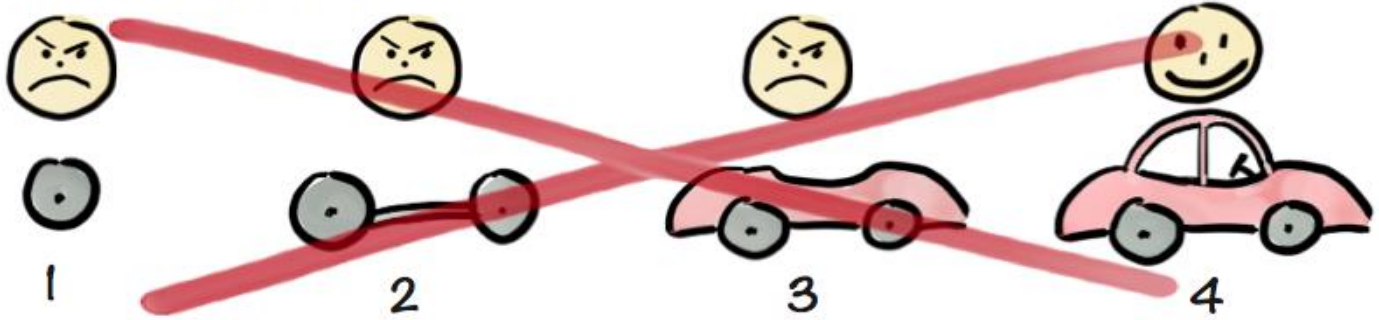
An iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams within an effective governance framework with “just enough” ceremony that produces high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders.

Scott Ambler

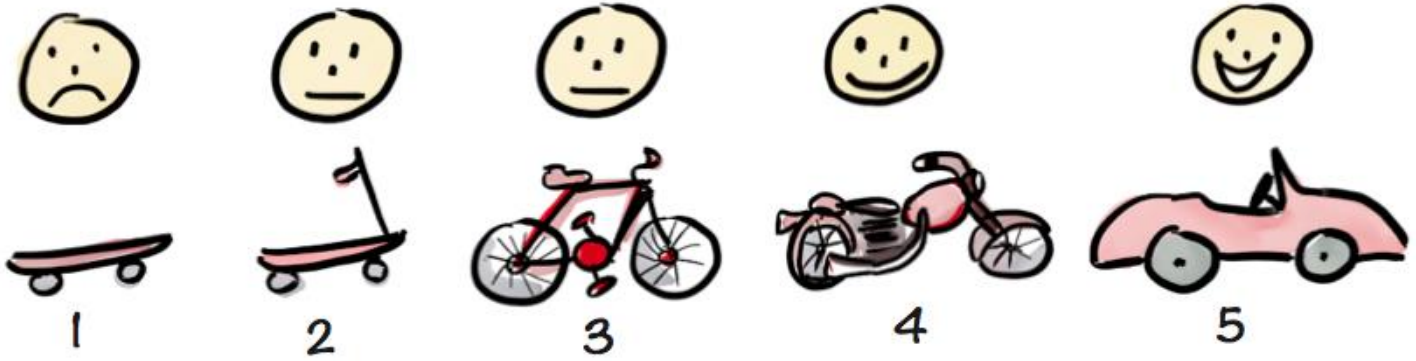
http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm

Agile software development describes a set of values and principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams. It advocates adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change. These principles support the definition and continuing evolution of many software development methods.

Not like this....



Like this!



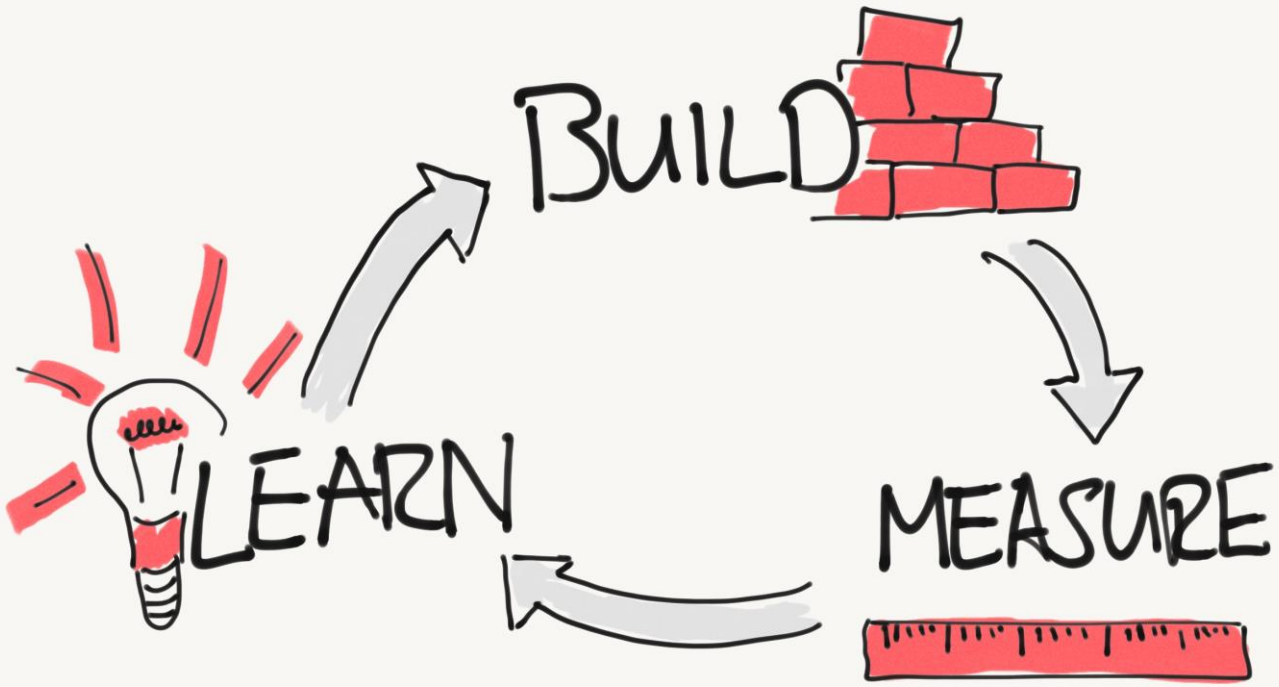
Henrik Kniberg

WHY AGILITY

- Need to more effectively respond to change – Organizational needs – Market demands – Threats and opportunities

- Manage evolutionary change to culture, products, and processes

- Frequently and incrementally deliver value to reach a desired goal or outcome


BUILD
MEASURE
LEARN

Agility is the ability to both create and respond to change in order to profit in a turbulent business environment

Agility is the ability to balance flexibility and stability

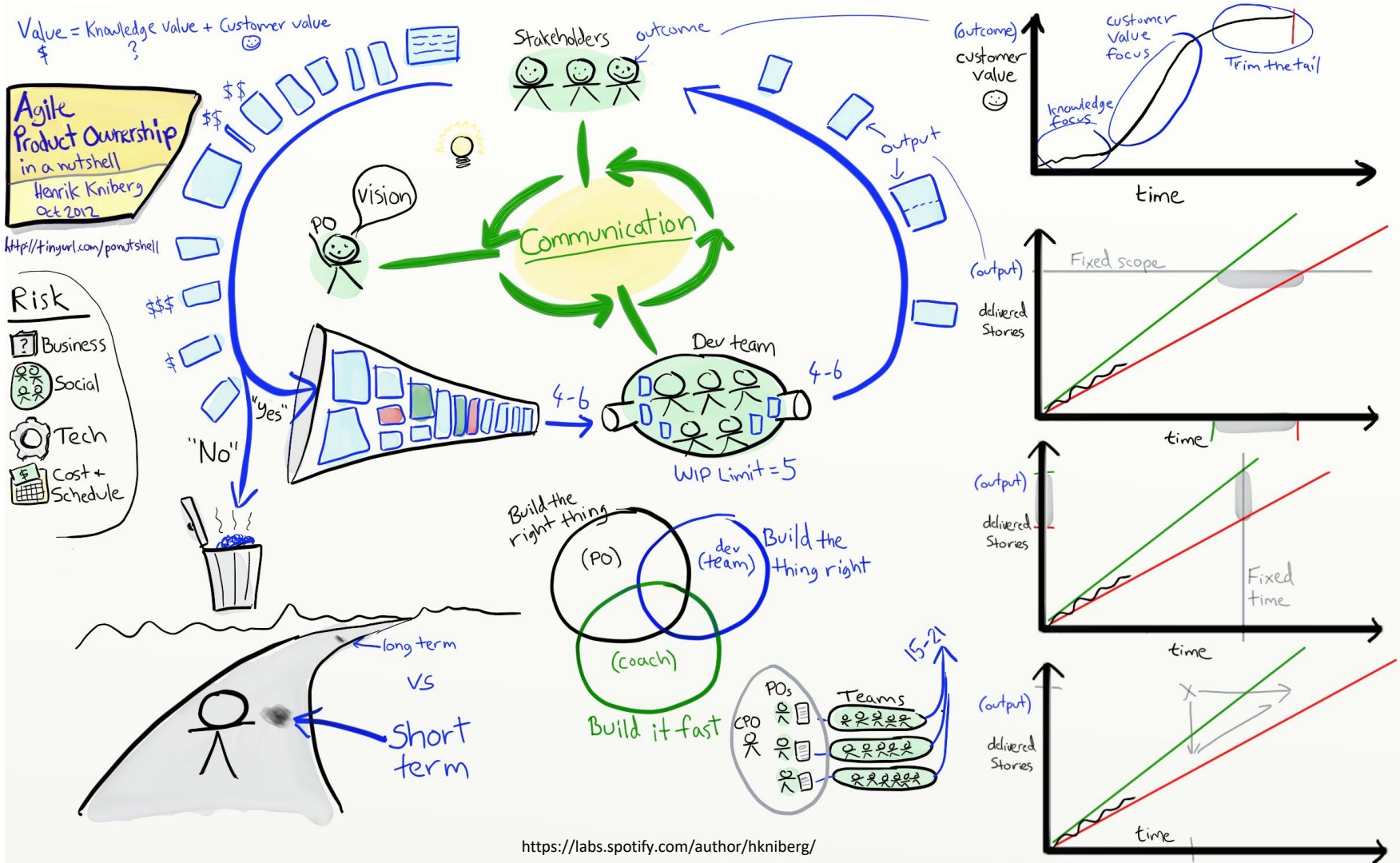
Jim Highsmith Agile Project Management (2nd Edition, 2010)

BENEFITS

- Greater responsiveness
- Measured increase in productivity
- Lower costs
- Managed risk through greater visibility
- Increased customer satisfaction
- Better overall quality
- Improved team morale

Agile principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity - the art of maximizing the amount of work not done - is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly



https://labs.spotify.com/author/hkniberg/

DO’S AND DON’TS

Agile methods do:
- Position us to be more responsive to changing priorities
- Allow better predictability of output for the foreseeable time horizon
- Make priorities and risk calculations more transparent

Agile methods don’t:
- Make engineers write more lines of code per day or designers create output faster
- Preclude the need to make good organizational decisions about product, priorities, and customer needs
- Allow the rest of the organization to assume they think they know what lies months ahead

In software development, sadly if you specify something, and everyone is doing their best, you’ll get what you want – at least what you specified. But, is it what you need?

AGILE MYTHS

- “Agile is the silver bullet for our organization!”
- “Agile doesn’t scale, so it can’t work for our organization!”
- “No more documentation or deadlines!”
- “Since we don’t do things upfront, we have no design and poor architecture!”
- “With so much rework going on, nothing new, and innovative gets worked on!”
- “Agile makes us faster!”

http://jpattonassociates.com/dont_know_what_i_want/

Perhaps you've been on this Agile project:
Customers meet with the team and successfully write a number of user stories. After a lot of conversation between developers and customers, developers estimate the stories. Customers prioritize them, highest value first, and choose the most important stories for the first release scheduled after six iterations.
Development starts, and things seem to go very well. In the fantasy world this story occurs in, all the development estimates were accurate. In the first couple iterations all scheduled stories are finished. But, that's where things go wrong.
After looking at the resulting software the customer says "Now that I see this, we're missing a few things. And, although the things you've built meet the acceptance criteria, we, well.. uh… weren't really sure about that acceptance criteria and now that we see it, it needs to change."
"No problem" says the team. "Just write more stories. But, you'll have to remove some of the others from this release in order to get them done on time."
The customer's shocked and angry. "What you're saying is that I needed to get the requirements right up front! This smells just like waterfall – except without the up front time I'd need to even try to get the requirements right in the first place."

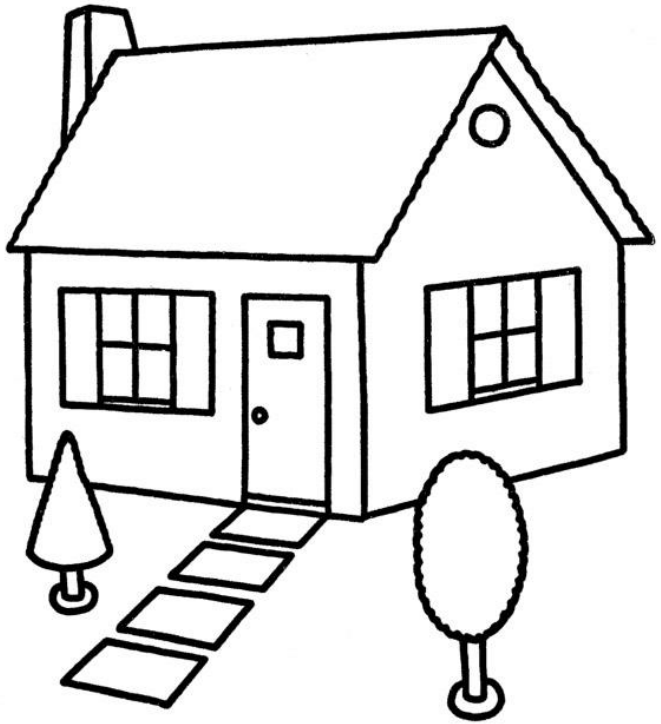
http://alistair.cockburn.us/Three+cards+for+user+rights

http://jpattonassociates.com/dont_know_what_i_want/

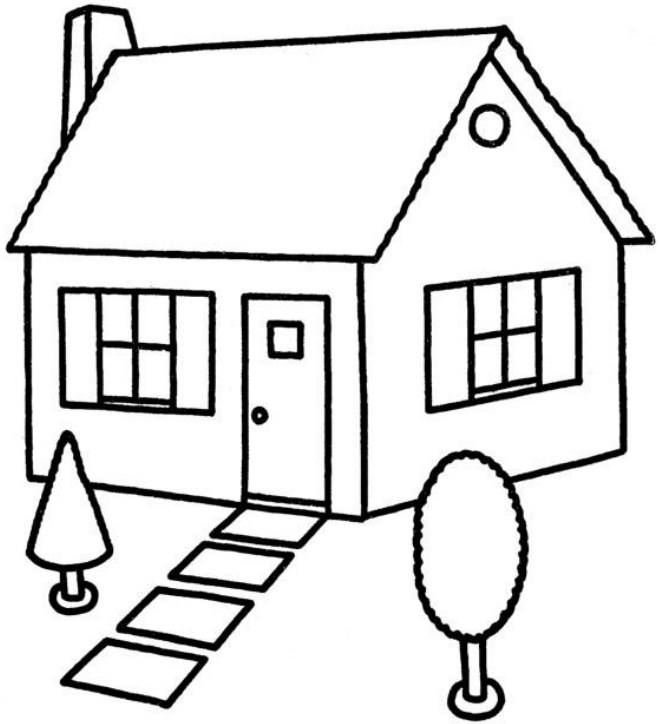
Agile **issues**

I was visiting a relative a couple years ago. My poor cousin (the CEO of an insurance company) had been sold the Agile Silver Bullet™ and was pissed. He said something like:
It’s a sham! We changed the way we do everything. We brought in consultants. We hired these master project managers. And nothing worked! It made no difference. There’s no accountability. All I get is excuses.

https://hackernoon.com/why-isnt-agile-working-d7127af1c552



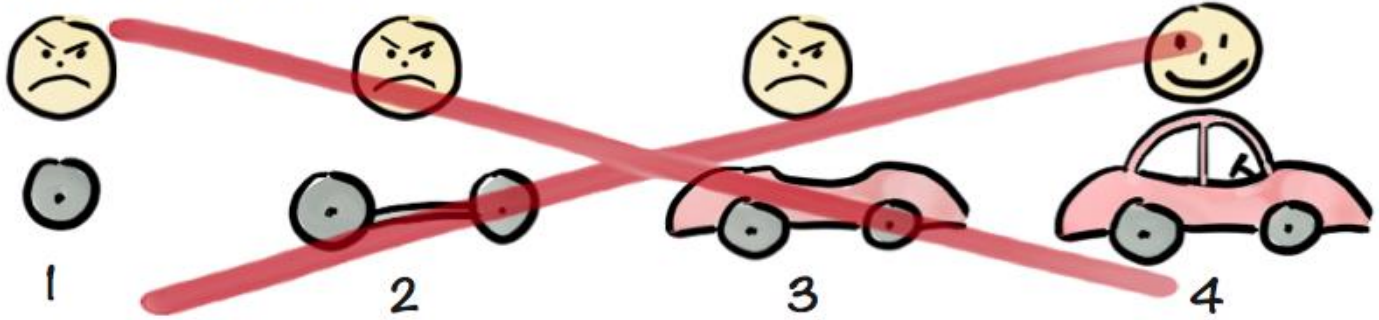
Walls
Windows
Doors
Roof
Chimney
Lawn
Swimming pool



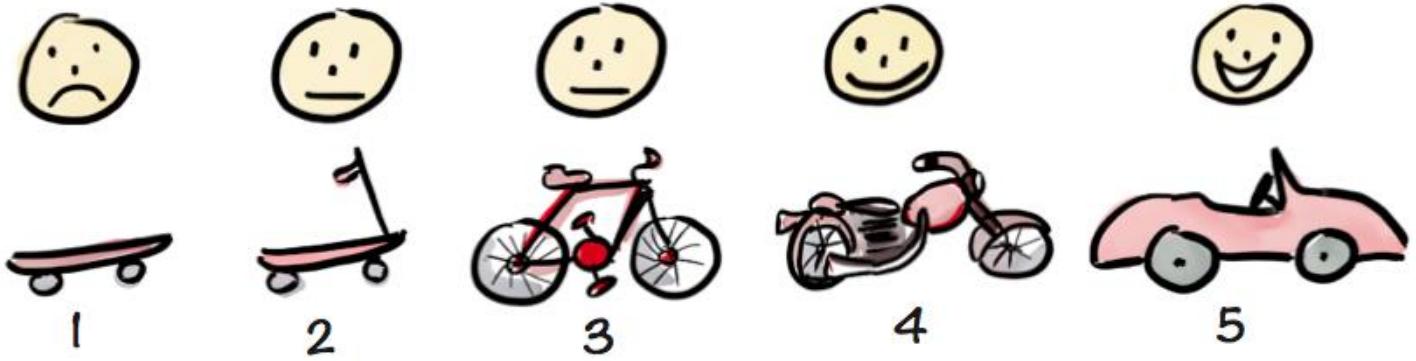
	STORY	VALUE
BUILD ROOF	34	144
WIRING	21	55
BUILD WALLS	55	89
HEATING	13	21
PLUMBING	34	55
CARPETS	2	2
DECORATE	8	2
SWIMMING POOL	55	1

This story needs to go first

Not like this....



Like this!

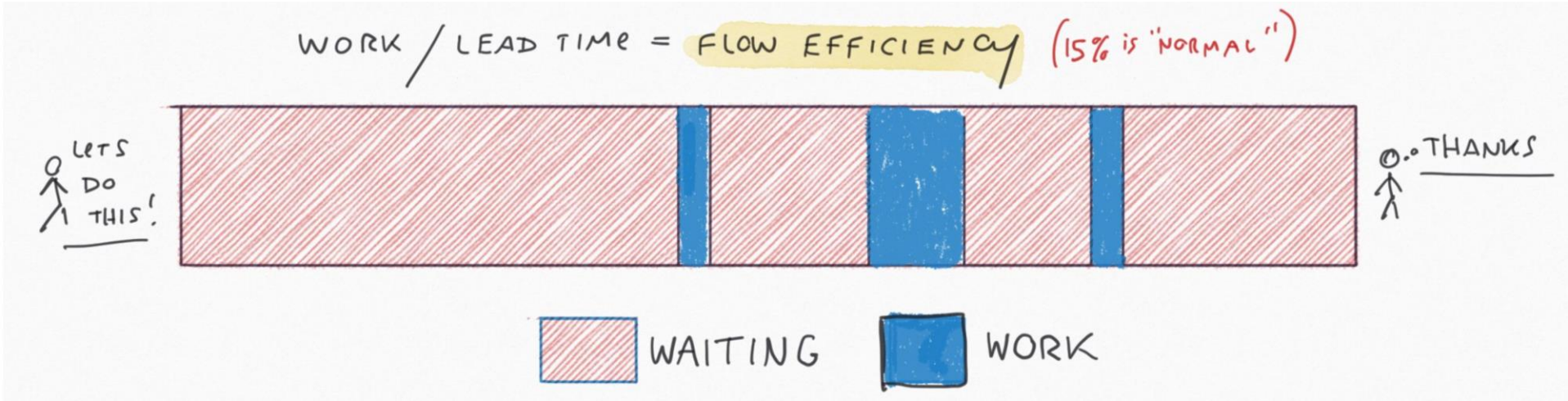


Henrik Kniberg

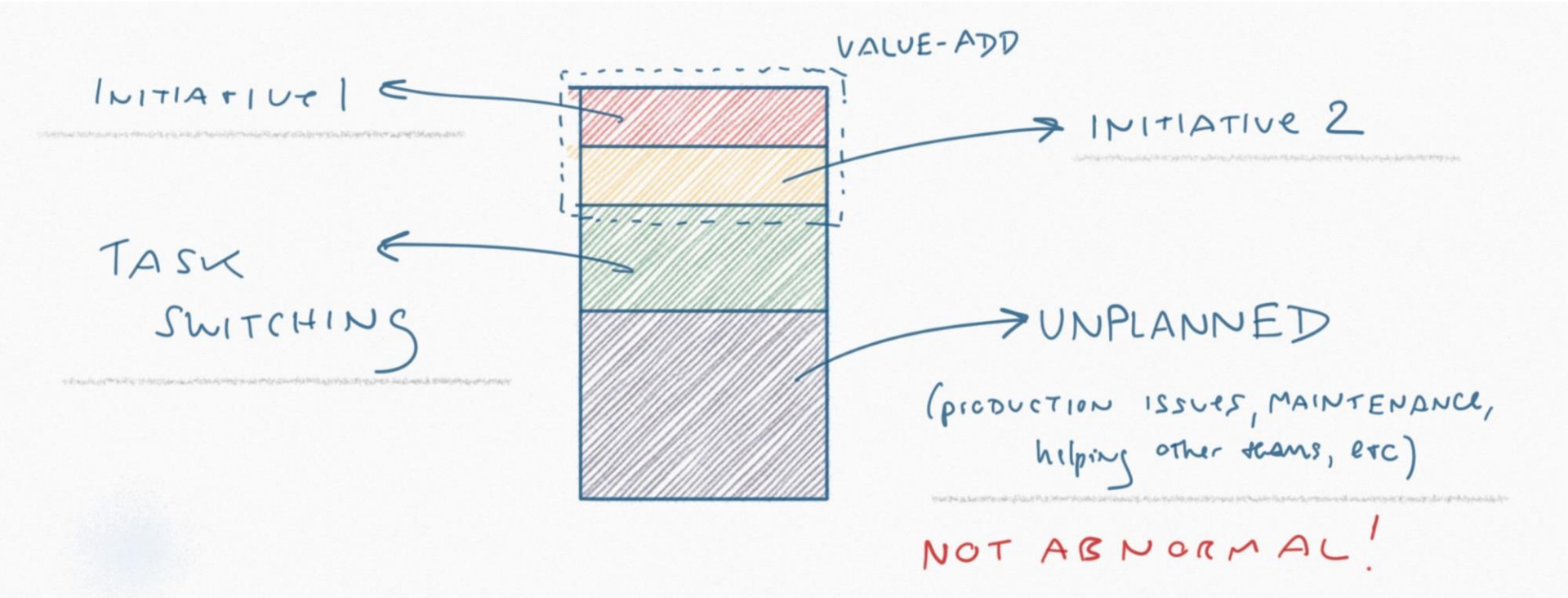

Would you drive a car on the road
… without proper training?
… without tyres?
WALLPAPERSWIDE.COM

Is not iterating

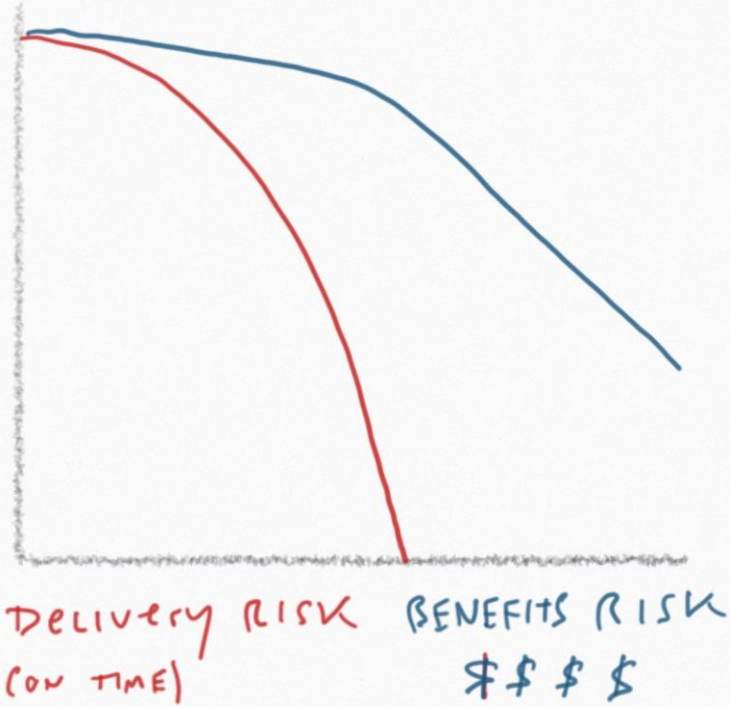
Flow effiecency

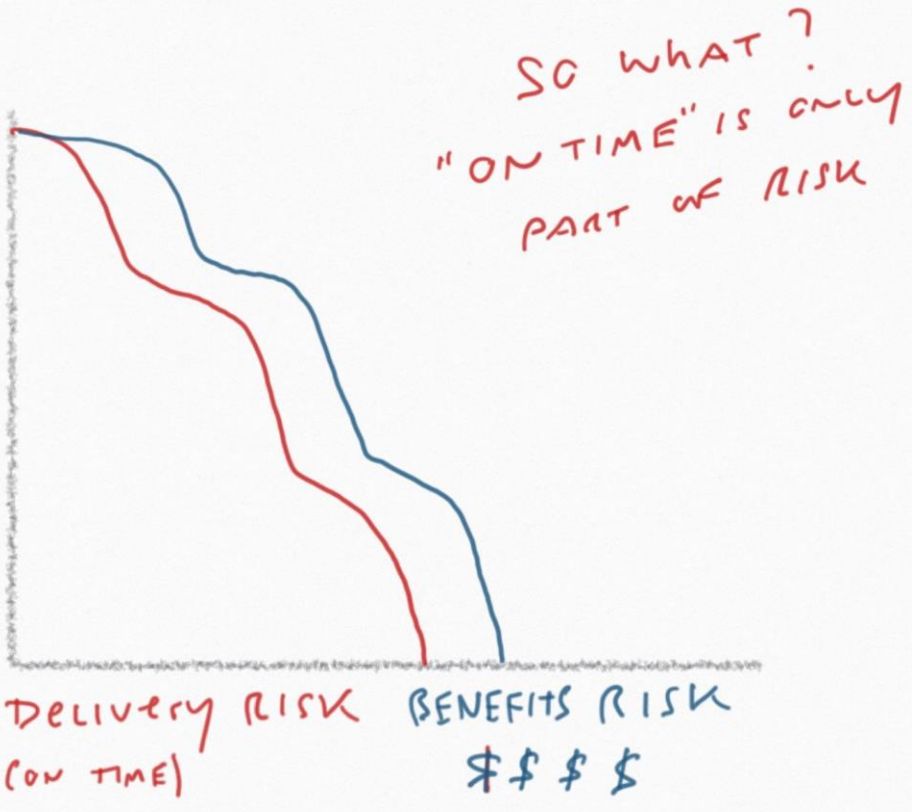


Unplanned work



Benefits realization





No learning / improvement

continuous improvement

Value and Waste

In assessing a process, it is important to understand what activities in the process actually add value to the end result. All other activities are wasteful.

CVA (Customer Value Added – or just VA for Value Added): adding form fit or function to a product or service, an activity that the customer would be willing to pay for in isolation if they knew it was being done – e.g. Creating code, implementing functionality.

BVA (Business Value Added – non-negotiable waste): an activity that is required to operate the business but the customer is unwilling to pay for – e.g. Budget tracking, code documentation.

NVA (Non-Value Added): an activity that is not required by the business nor is the customer willing to pay for – e.g. Waiting for resource allocation, requirements documents, estimation

What’s the First Thing the Team Could Do?

I like to ask, “How little can we do?”
Too often, the team has been asking “How much can we do?”

Sometimes, when we have stories, and we don’t know how to break them apart, we ask, “What’s the first thing that could add value?” Maybe that’s a good question here. You could ask, “What’s the first thing the team can do?” Or, “What’s the first thing we can do that would help us get to done in a one- or two-week iteration?” Or, “How do we finish a feature

The point of using agile is to get finish something valuable-to-the-business quickly, to get feedback. Why? For several reasons, but the first one is so you can change the project's priorities. The second is so you can change the project portfolio. The third is to get feedback on what you've done. This is why every project needs to design its own way to get to done.

What happens when you have more unknowns? What if your organization is "addicted" to waterfall and long projects? When you can't see the people on your project, because everyone isn't in the same place? When you have a lot of technical risk, such as technical debt? How about when you're starting a program and everyone is transitioning to agile (oh please, don't do this). Or, you're new to agile, and you don't have training. I have a question: would you drive a car on the road with no training, either?

Definition of Done

When are you done with a feature?

Getting the users' feedback incorporated into the feature is part of the basic cost of developing the feature.

| Started | In progress | Done |

| Started | Ready for user review 1 | Ready for user review 2 | Ready for user review 3 | Accepted |

Selecting development strategies

- Some people like to get each story **as correct as possible** on the first round, and try to **minimize the changes** in rounds 2 and 3.
- Some people like to do the **minimum possible** on the first round, so they have the **most time to steer** according to the review feedback.

Acceptance criteria

http://alistair.cockburn.us/Three+cards+for+user+rights

← SPRINT 7 09/01/13 → 30/01/13
STORIES
7 WEEKS TO RELEASE (TARGET: 28th Feb)
TO DO
NEXT
IN PROGRESS
DONE
IN TEST
URGENT! (max 1)
BUGS / SUPPORT
ASYNC
STARTUP
"Other Stuff" (technical)
DEV
UX / TC
TEST

Days left

12

Factors For Successful Projects

User involvement
Executive management support
Clear statement of requirements
Proper planning
Realistic expectations
Smaller project milestones
Competent staff ownership
Clear vision & objectives
Hard-working, focused team

Engineering

Needs to have constant learning, refinement and adaptation to meet the environmental requirements

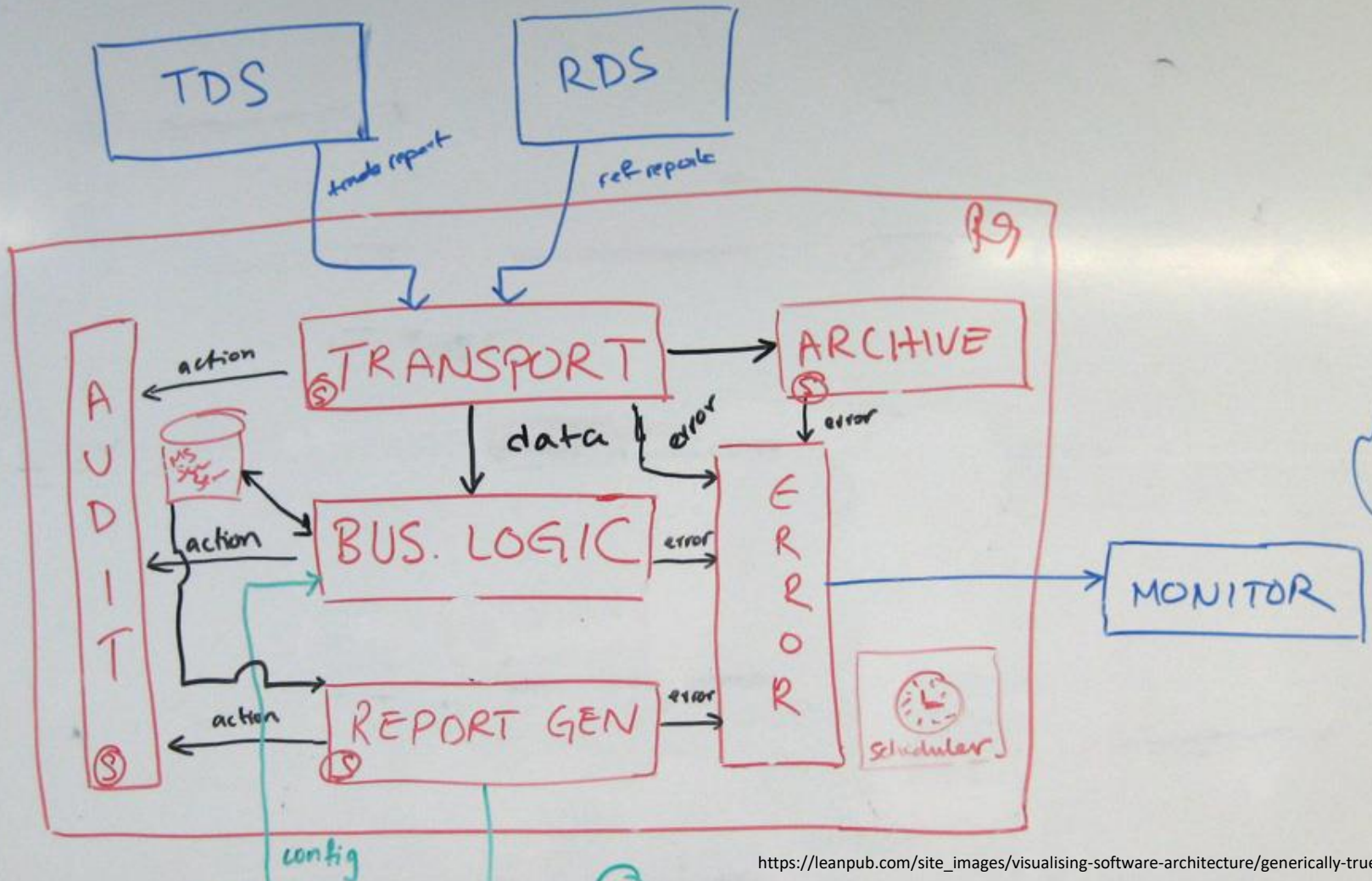
Architecture

What is software architecture?

Bass, Clements, and Kazman:

“

The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.

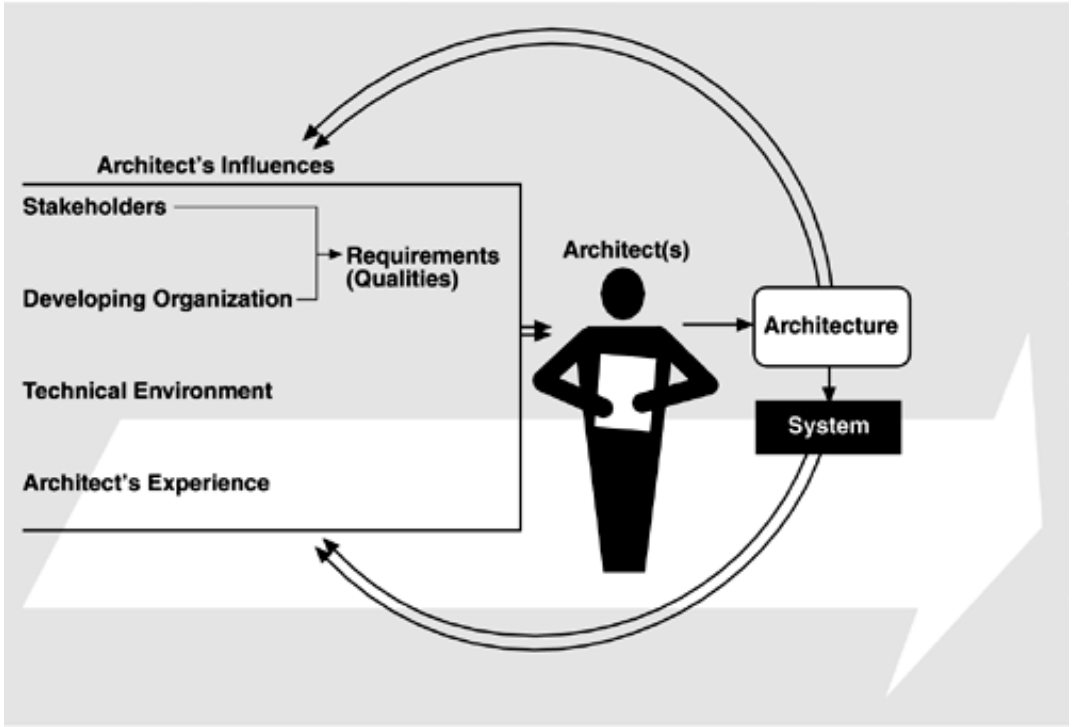


https://leanpub.com/site_images/visualising-software-architecture/generically-true.jpg

Software architecture is a result of

business, technical & social

influences

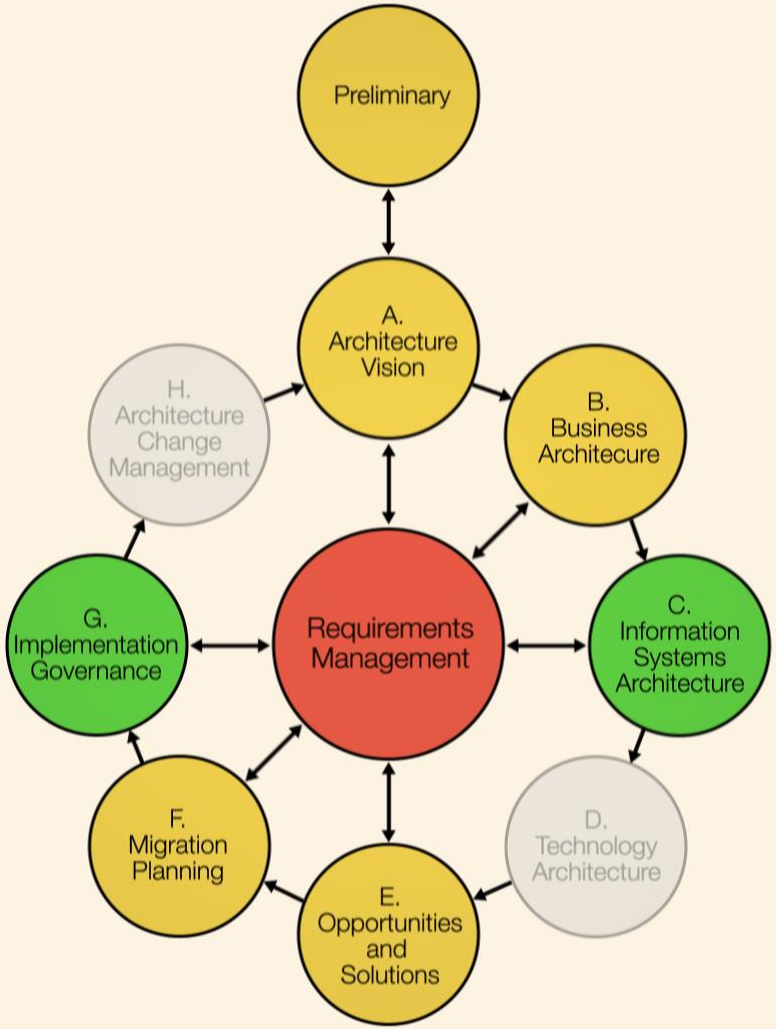


Questions

1. How do requirements lead to an architecture?
2. How do you analyze architectures?
3. How does architectures enable new organizational requirements and needs?
4. Is the architecture slowing you down?

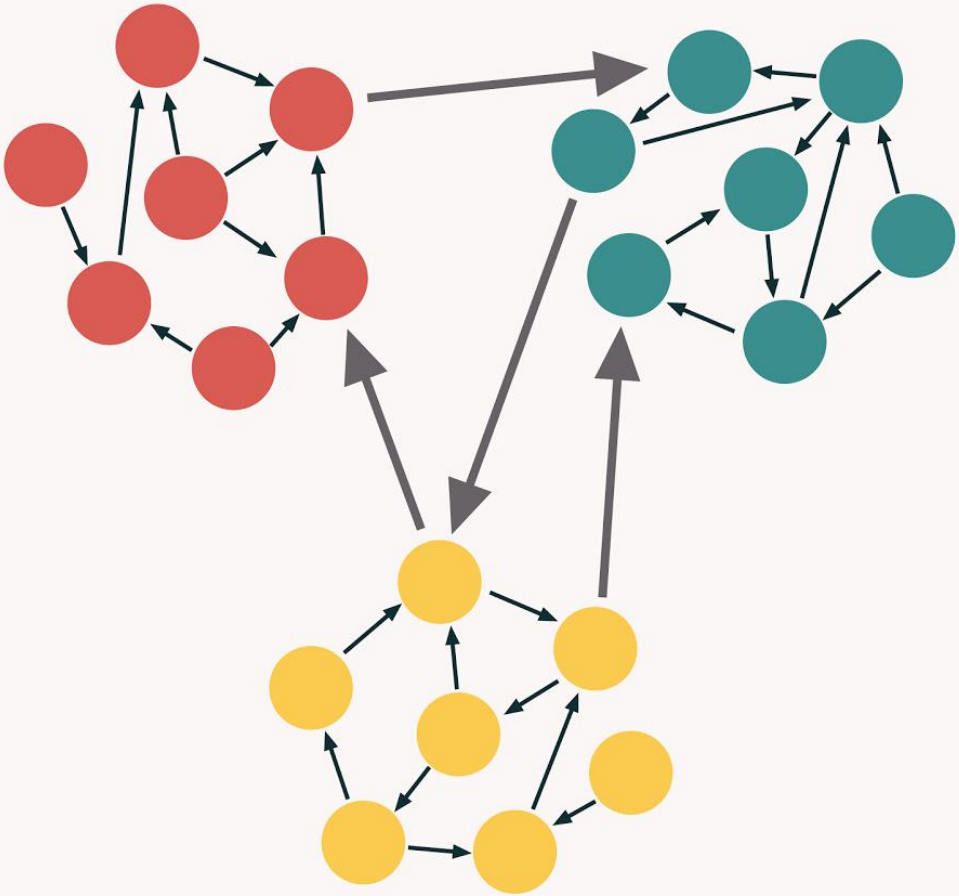
Why software architecture?

1. Enable others to understand the system
2. Allow others to effectively work on parts of the system in isolation
3. Allows for extensions of the system
4. Facilitate reuse and reusability
5. Based on good principles and practices



ARCHITECTURE
DEVELOPMENT
METHOD

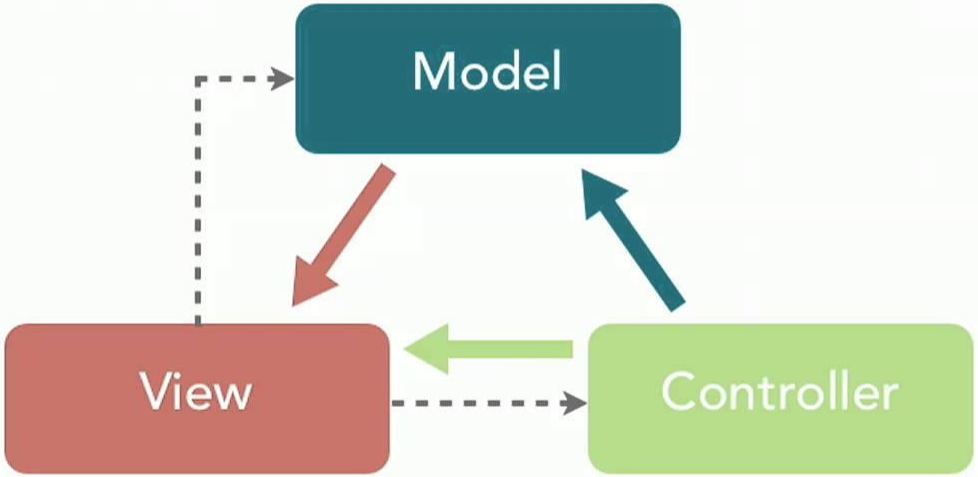
to
meet
your
specific
needs



High cohesion

Loose coupling

the Model-View-Controller architecture pattern

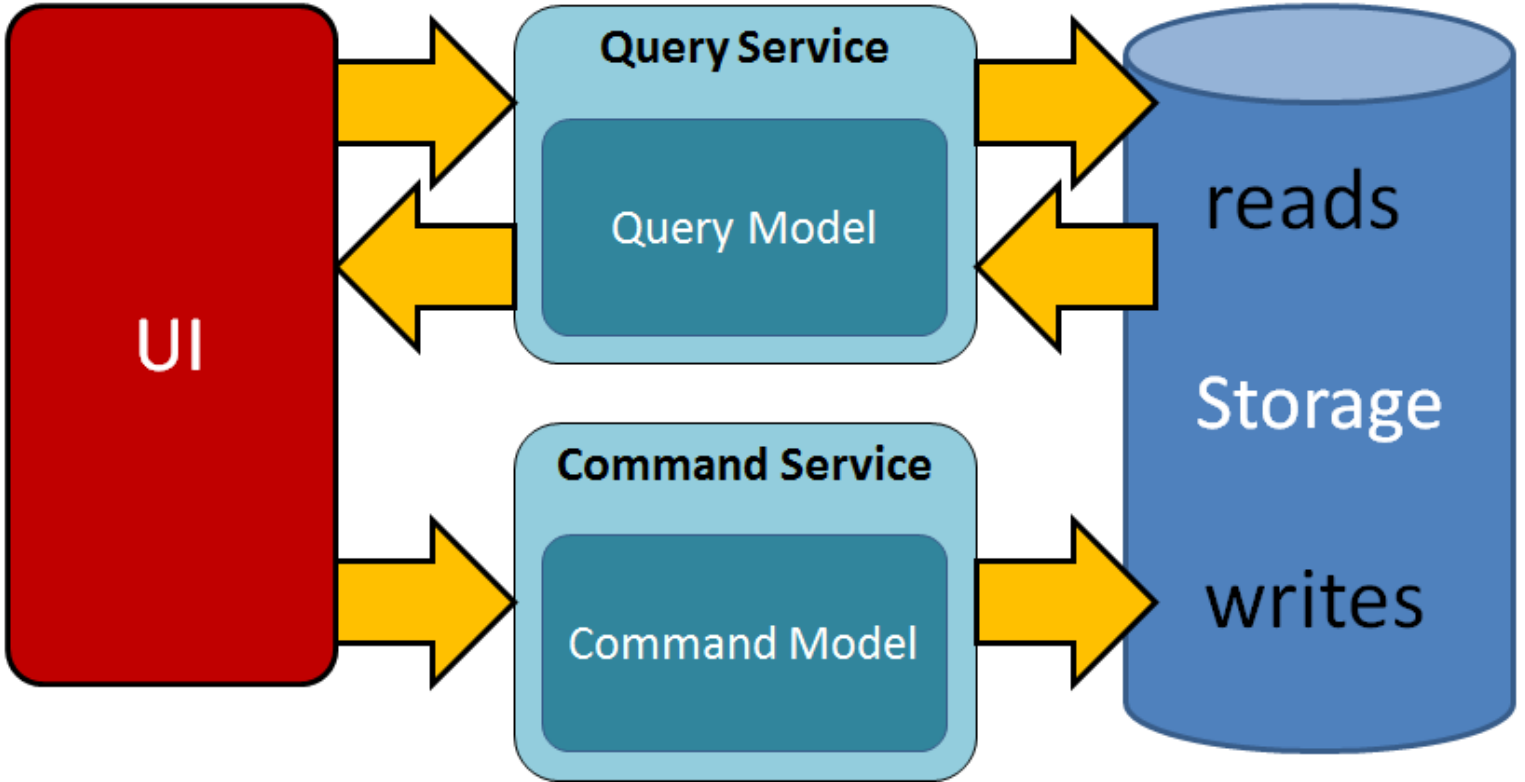


Model
contains business logic

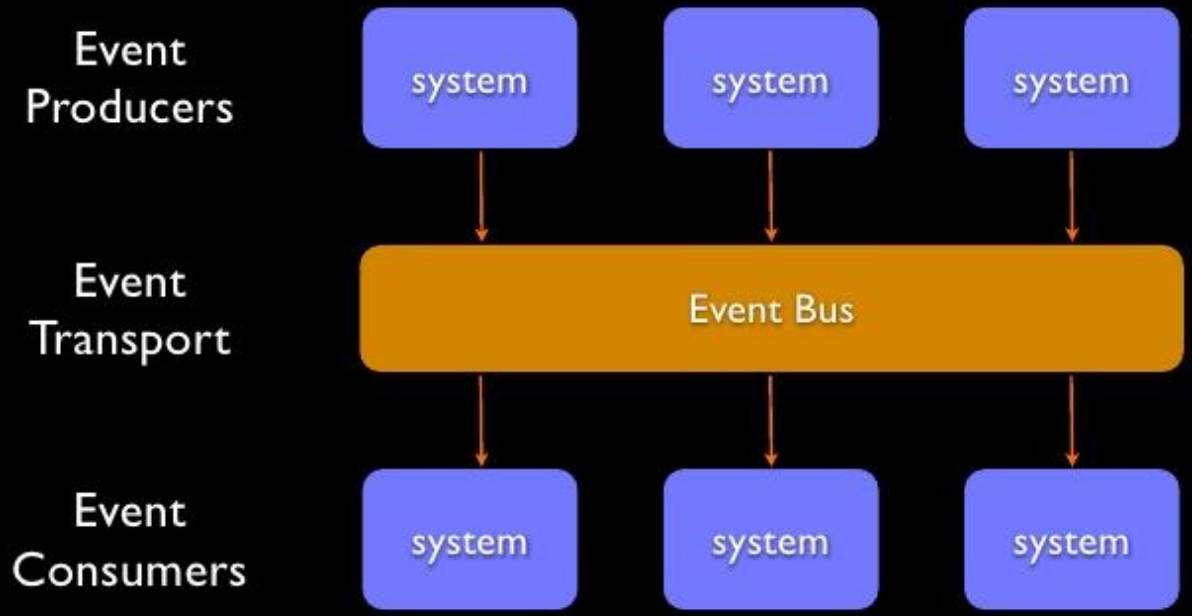
Controller
interacts with **Model** to create data for the **View**

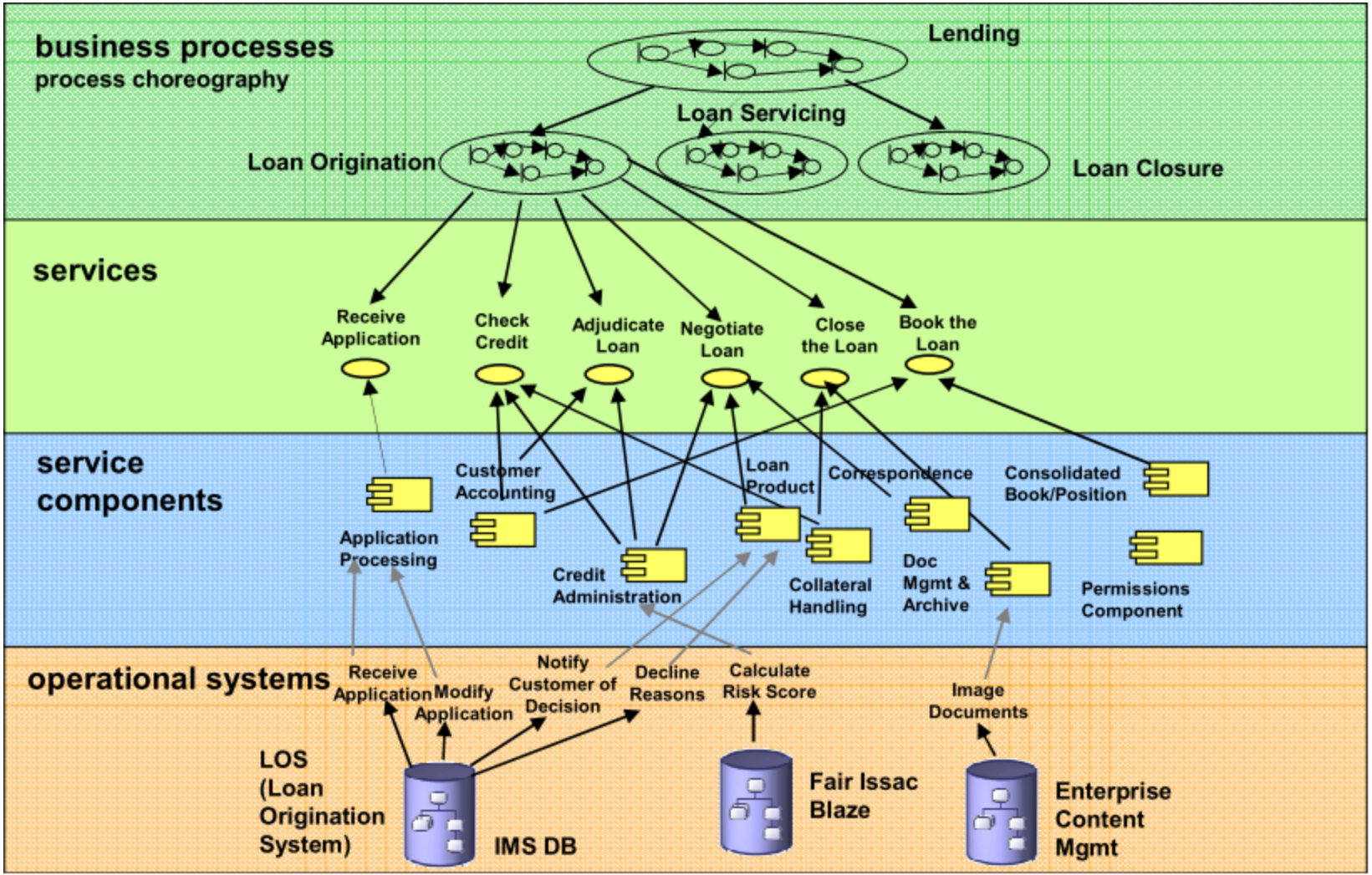
View
renders content to the user and relays user commands to the **Controller**

CQRS Pattern



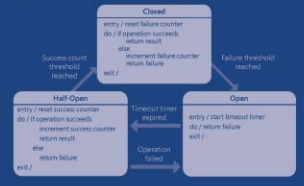
Typical EDA architecture




business processes
process choreography
Lending
Loan Servicing
Loan Origination
Loan Closure
services
Receive Application
Check Credit
Adjudicate Loan
Negotiate Loan
Close the Loan
Book the Loan
service components
Application Processing
Customer Accounting
Credit Administration
Loan Product
Collateral Handling
Correspondence
Doc Mgmt & Archive
Consolidated Book/Position
Permissions Component
operational systems
Receive Application
Modify Application
Notify Customer of Decision
Decline Reasons
Calculate Risk Score
Image Documents
LOS (Loan Origination System)
IMS DB
Fair Issac Blaze
Enterprise Content Mgmt

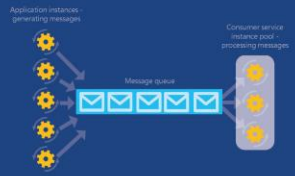
Cache-aside

Circuit Breaker

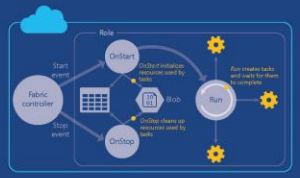


Compensating Transaction

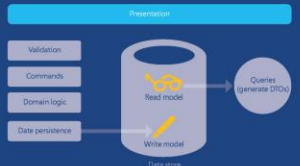
Competing Consumers



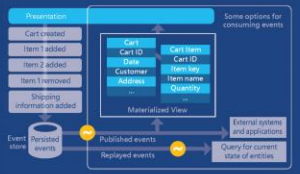
Compute Resource Consolidation



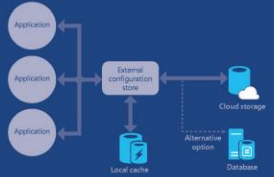
Command and Query Responsibility Segregation (CQRS)



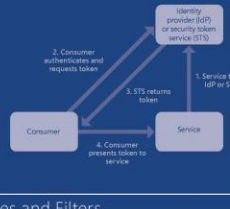
Event Sourcing



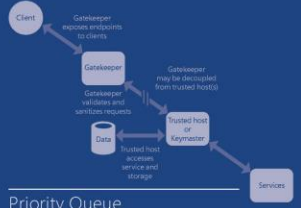
External Configuration Store



Federated Identity

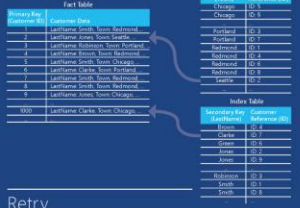


Gatekeeper

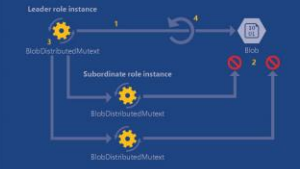


Health Endpoint Monitoring

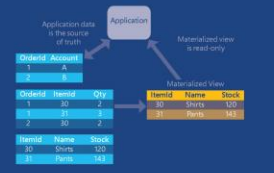
Index Table



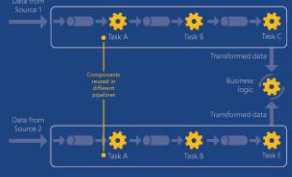
Leader Election



Materialized View

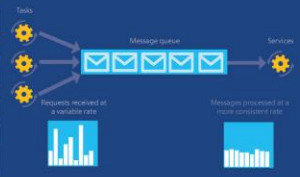


Pipes and Filters



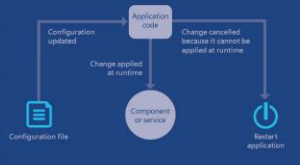
Priority Queue

Queue-Based Load Leveling

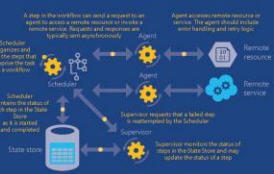


Retry

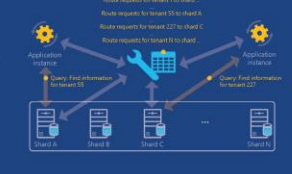
Runtime Reconfiguration



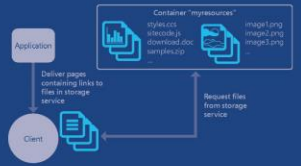
Scheduler Agent Supervisor



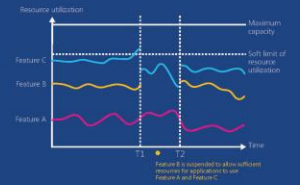
Sharding



Static Content Hosting



Throttling



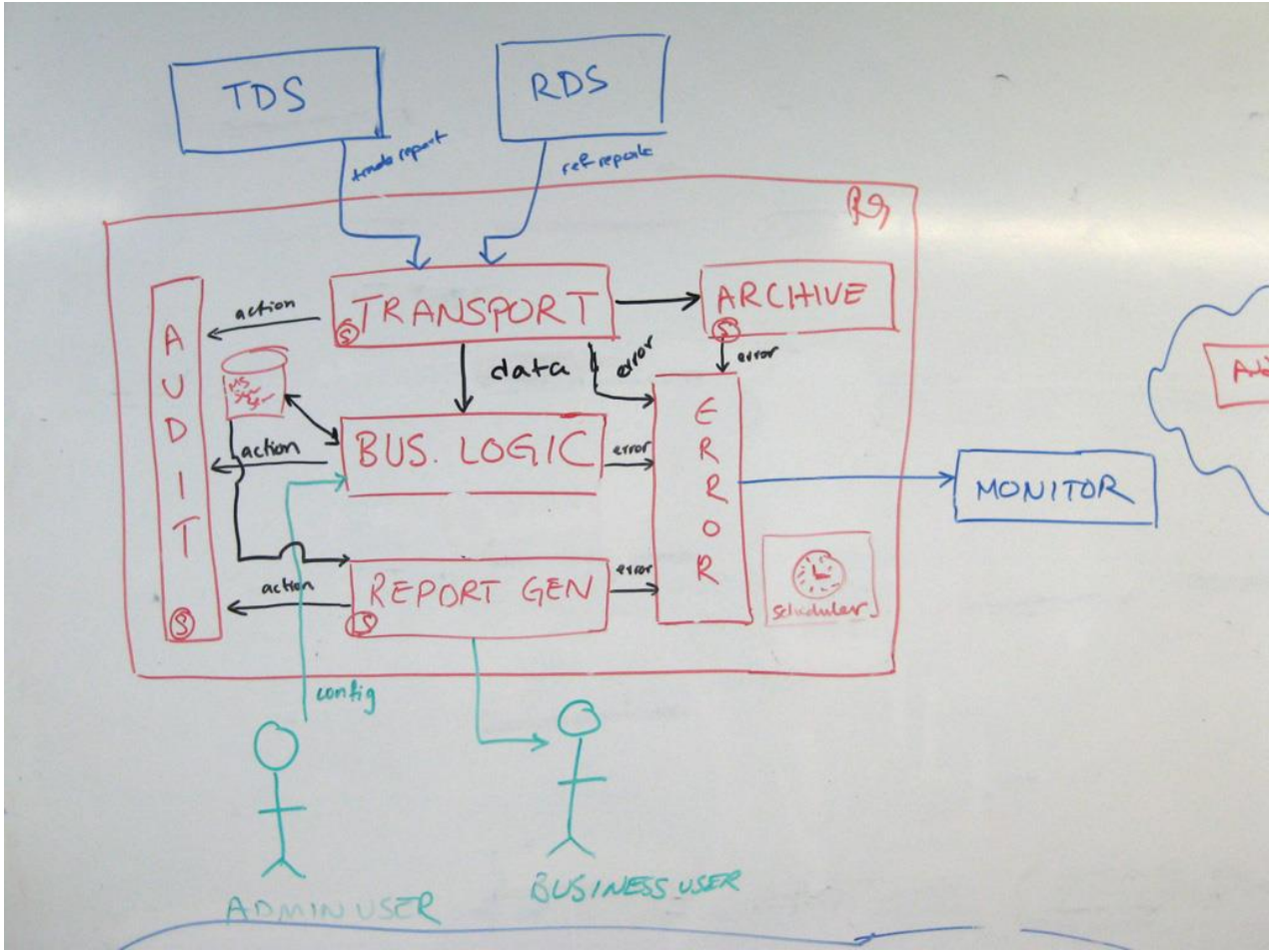
Valet Key

What do we need to ask ourselves

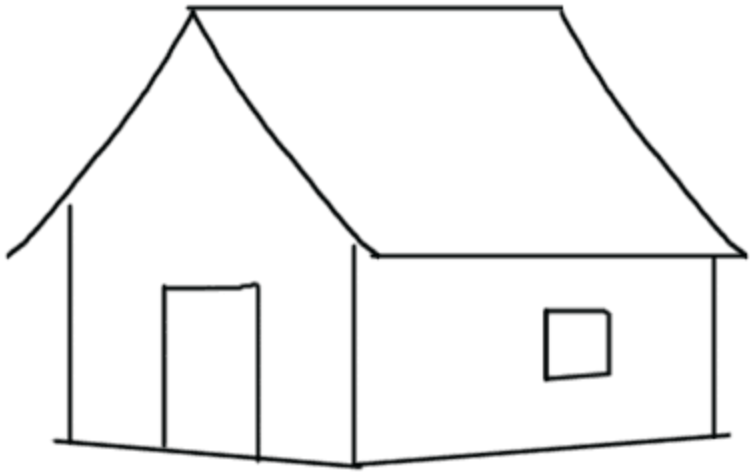
1. Does the architecture fulfill the business goal?
2. What is the purpose of a diagram and who is the recieving end?
3. Does the architecture exhibit high cohesion and low coupling?
4. What known design patterns fit the problem at hand?
5. Does the architecture permit multiple teams to work on independent parts effiecently?
6. What positive and negative effects does the created architecture exhibit?

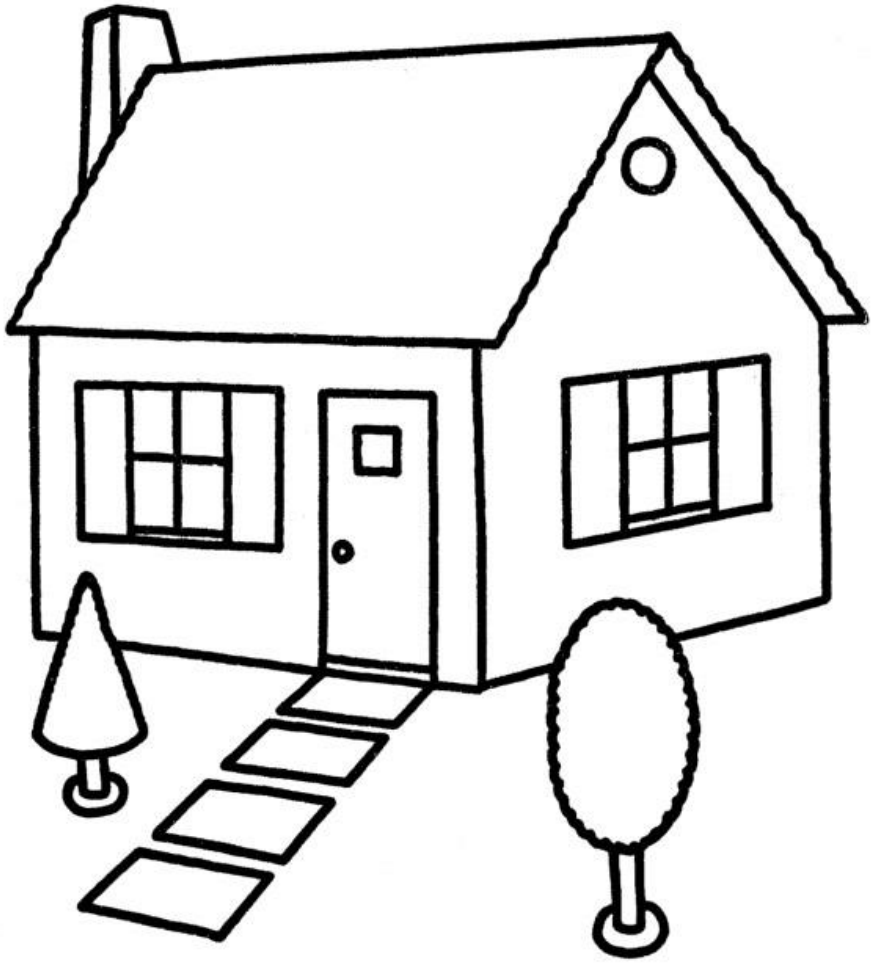

Architecture complexity

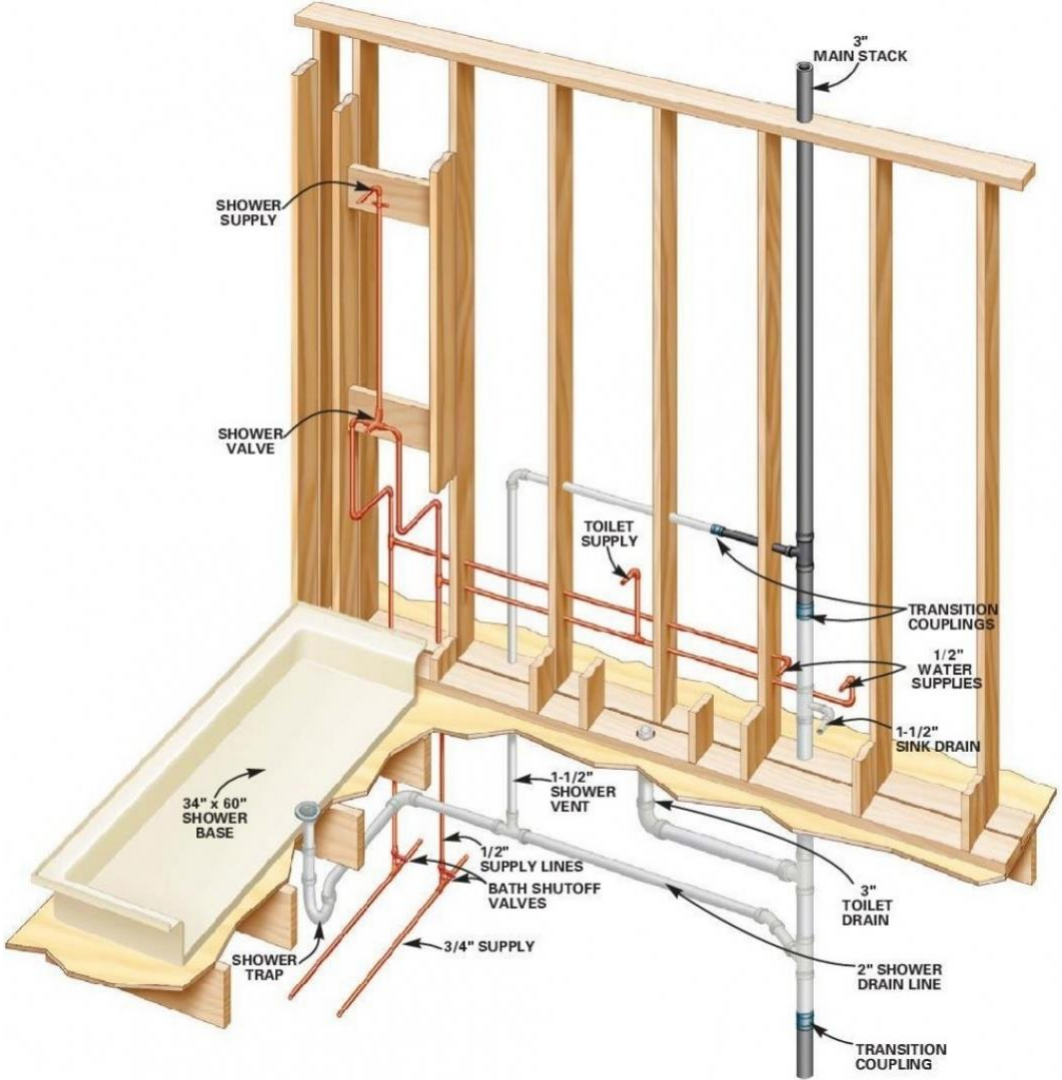

COMOYO



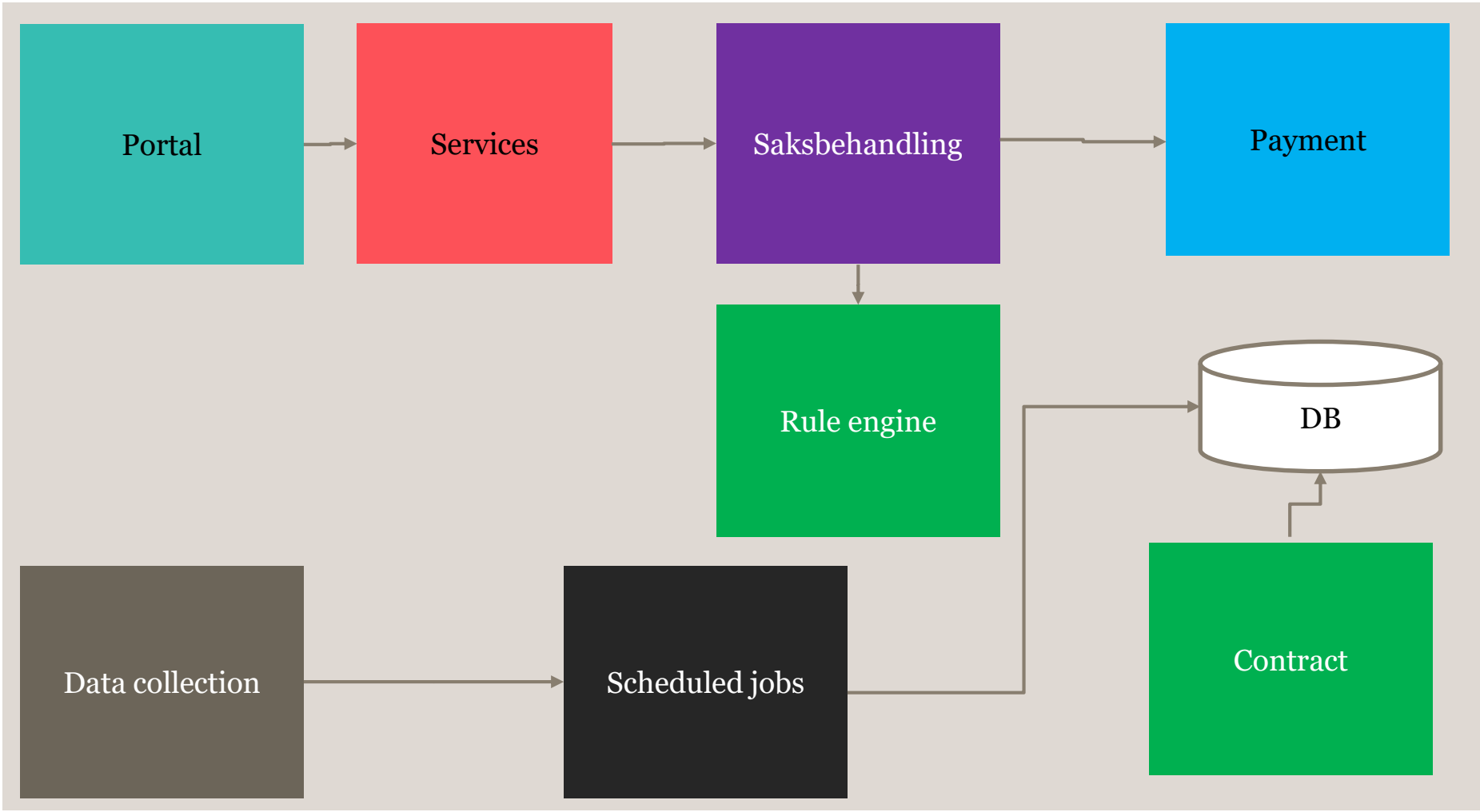
https://leanpub.com/site_images/visualising-software-architecture/generically-true.jpg



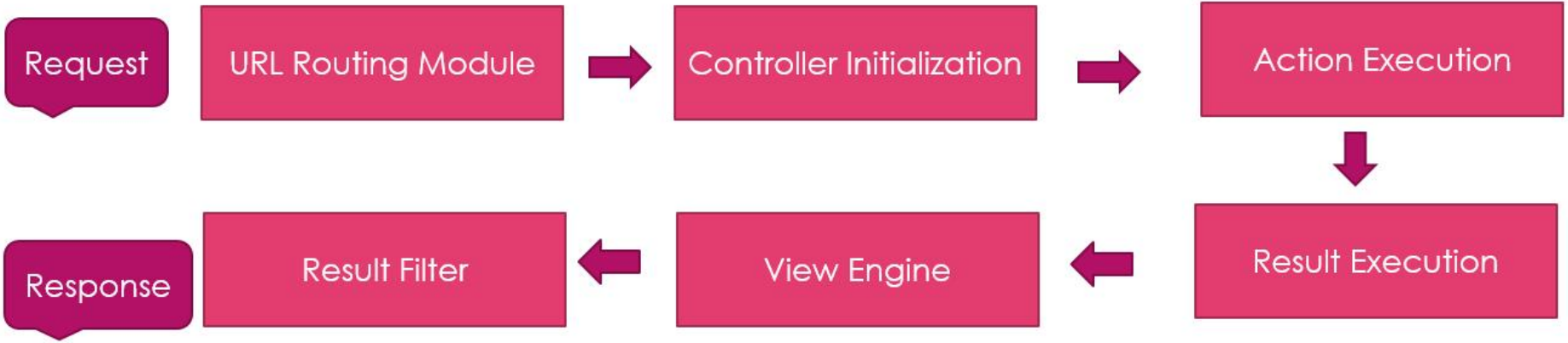


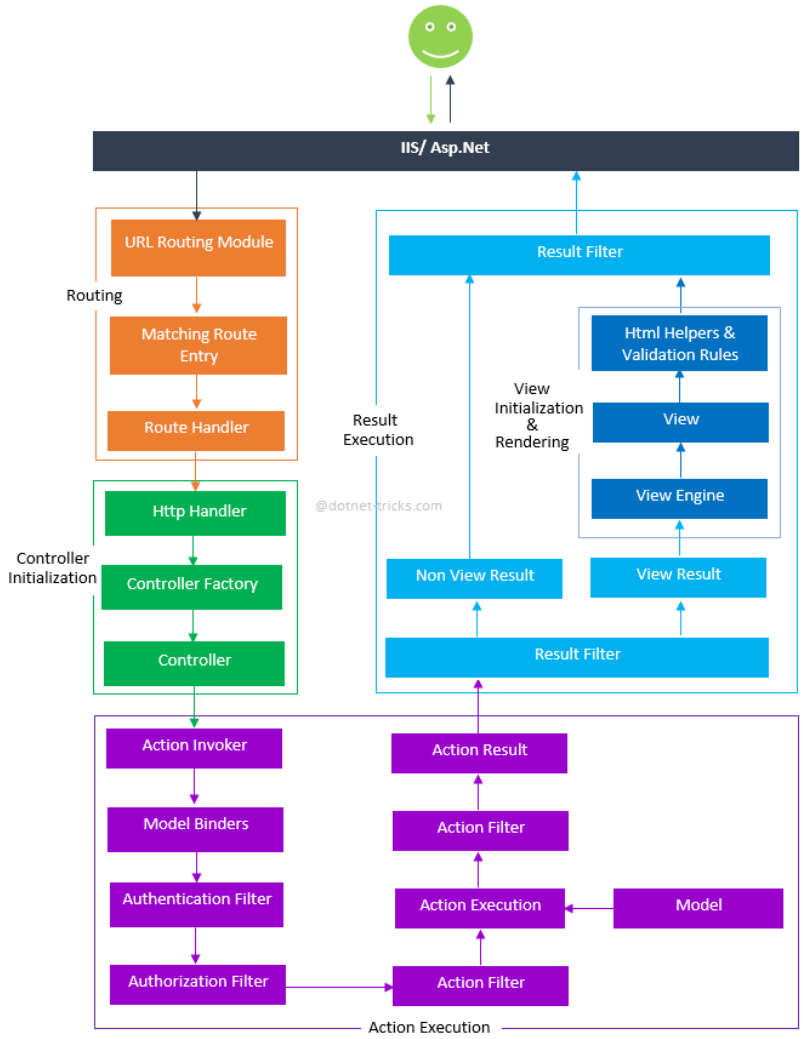

3"
MAIN STACK
SHOWER
SUPPLY
SHOWER
VALVE
TOILET
SUPPLY
TRANSITION
COUPLINGS
1/2"
WATER
SUPPLIES
1-1/2"
SINK DRAIN
34" x 60"
SHOWER
BASE
1-1/2"
SHOWER
VENT
1/2"
SUPPLY LINES
BATH SHUTOFF
VALVES
3"
TOILET
DRAIN
SHOWER
TRAP
3/4" SUPPLY
2" SHOWER
DRAIN LINE
TRANSITION
COUPLING

The system

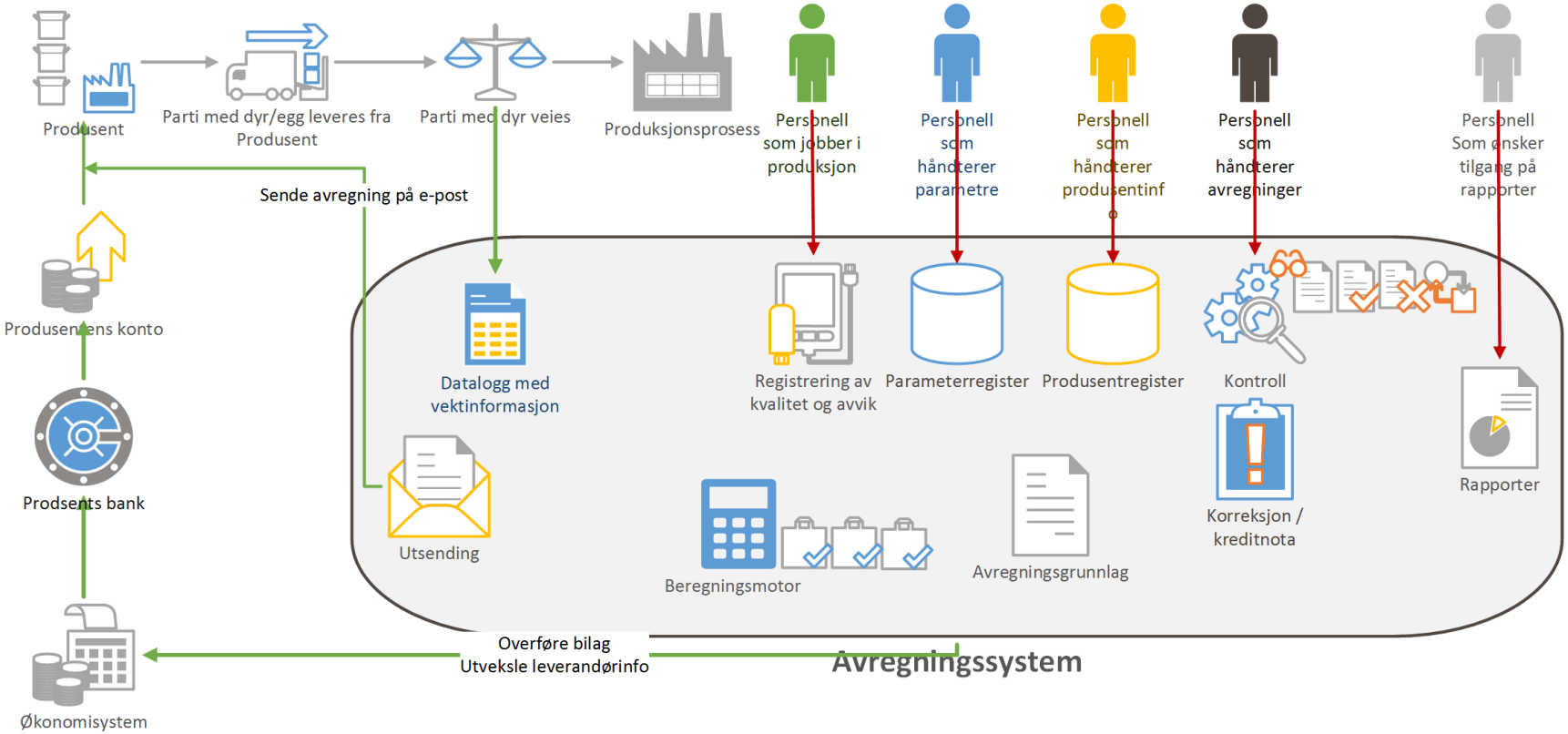

Portal
Services
Saksbehandling
Payment
Rule engine
DB
Data collection
Scheduled jobs
Contract

MVC FLOW


Request
URL Routing Module
Controller Initialization
Action Execution
Result Execution
View Engine
Result Filter
Response



ASP.NET MVC Pipeline


Produsent
Parti med dyr/egg leveres fra Produsent
Parti med dyr veies
Produksjonsprosess
Personell som jobber i produksjon
Personell som håndterer parametre
Personell som håndterer produsentinfo
Personell som håndterer avregninger
Personell Som ønsker tilgang på rapporter
Sende avregning på e-post
Produsentens konto
Prodsents bank
Økonomisystem
Datalogg med vektinformasjon
Utsending
Registrering av kvalitet og avvik
Parameterregister
Produsentregister
Kontroll
Korreksjon / kreditnota
Rapporter
Beregningsmotor
Avregningsgrunnlag
Overføre bilag
Utveksle leverandørinfo
Avregningssystem

Øker studiestøtten til elleve måneder

Regjeringen foreslår å utvide studiestøtten fra neste sommer. Det blir en ekstra måned med studielån, og totalt øker studiestøtten fra 94.000 kroner i år til 105.600 kroner. – Vi er veldig glade, sier NSO.

VELDIG FORNØYD: – Dette er et historisk gjennomslag for studentstøtten, sier leder i Norsk Studentorganisasjon Ola Magnussen Rydje.

Anne Lise Stranden

(DinePenger) Publisert: 11:34 - 14.10.2013, Oppdatert: 11:51 - 14.10.2013

En ekstra måned. Det legges opp til elleve måneder utdanningsstøtte per undervisningsår til studenter og andre som får støtte etter samme regelverk. Første ekstra måned med støtte vil bli juni 2015.

The Malleable Nature of Software

Evolution is more important in software than in other engineering disciplines

Software engineering rarely involves “green field” development. Software needs to be constantly maintained and evolved to meet new business requirements. The cost incurred in evolution usually exceed the development cost by a factor of 3 or 4

Conclusions?

The following criteria are evaluated in an integrated way:

- Whether the group has solved the given assignment, according to the customer's objectives of the project.
- Team work efficiency and the team dynamics
- Team work process improvement efforts
- Reasonable grounds for decisions taken.
- Visibility of limitations done.
- The students' ability to reflect on the process during the project.
- Layout and structure readability.
- Logical flow in the report.

Making software is not just about writing code.

It is equally much as to make informed decisions. This is why there are developed methods both in risk management, estimation, project management, architecture management.

Knowing about various principles, methodologies, patterns and good practices helps.

Think about how you can document and learn from your process.

Be aware of the choices you make – know why you make them

It takes a great deal of effort to make agile work.

Agile cannot function in all circumstances – it is your job to identify whether the organization is mature enough to

Communication is the most important factor

Project retrospective

Also known as post mortem reviews

A common method to share knowledge within a development team.

Safurudin Mahic

safurudin.mahic@bekk.no