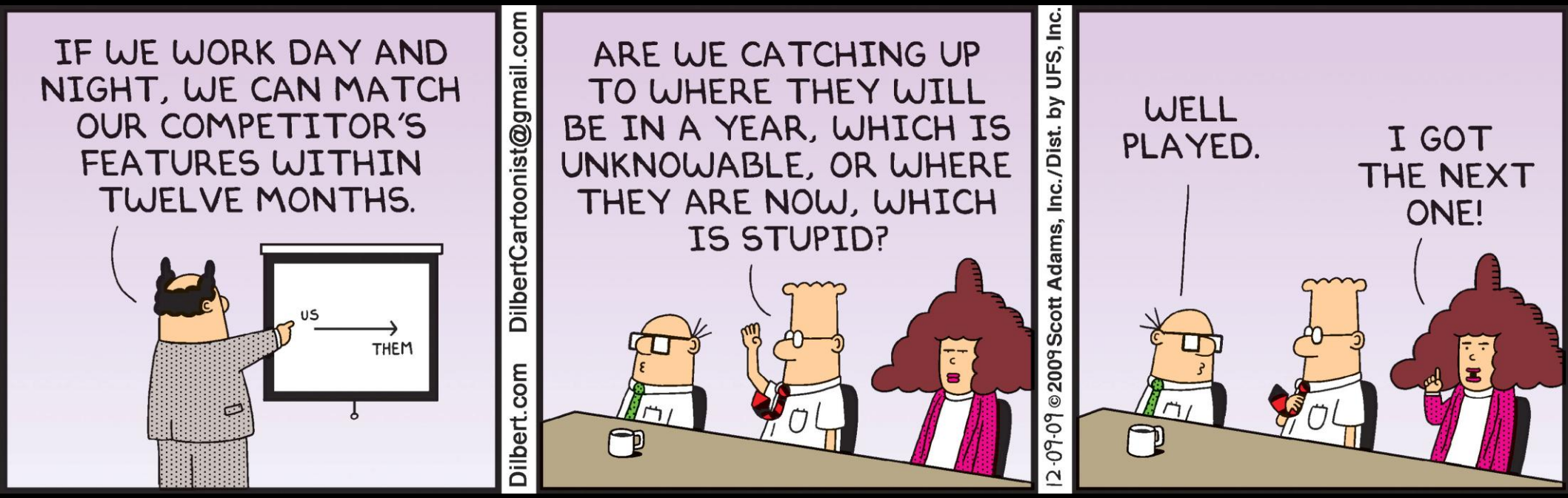
DAT255 / DIT543 SOFTWARE ENGINEERING PROJECT



https://github.com/hburden/DAT255/blob/master/README.md

TEACHERS

Håkan Burden

RISE Viktoria

burden@chalmers.se

Jan-Philipp Steghöfer

Chalmers | University of Gothenburg

jan-philipp.steghofer@cse.gu.se

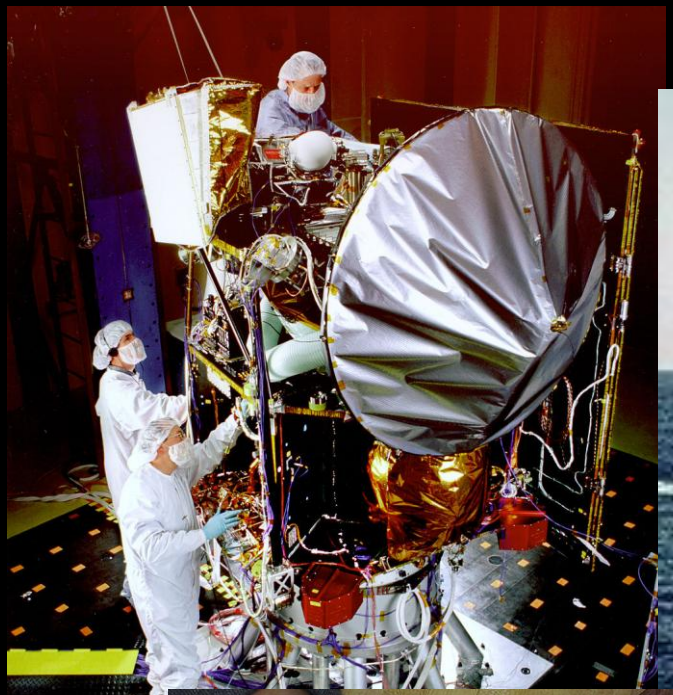
AGENDA

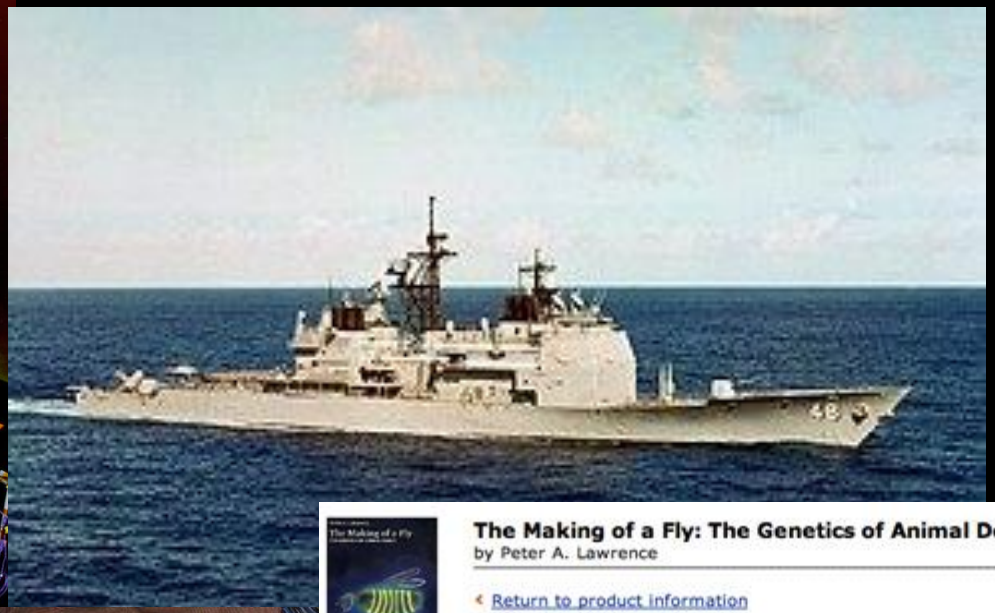
08-10

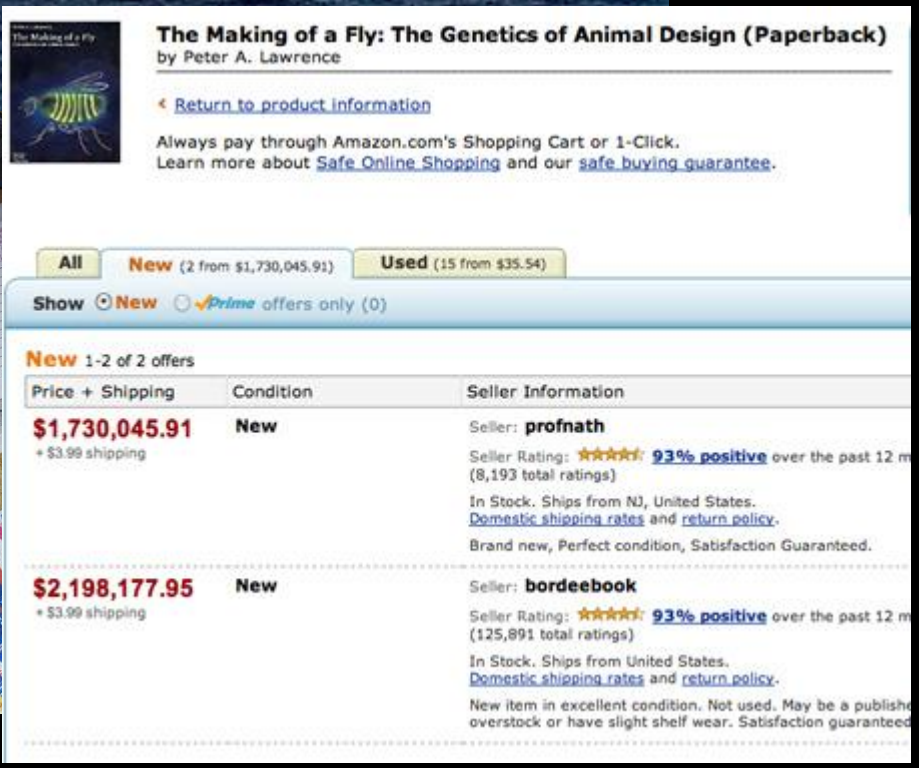
- Software Engineering
- Course details

10-12

- Kata exercise
- KPIs







The Making of a Fly: The Genetics of Animal Design (Paperback)
by Peter A. Lawrence

‹ Return to product information

Always pay through Amazon.com's Shopping Cart or 1-Click.
Learn more about Safe Online Shopping and our safe buying guarantee.

All | New (2 from $1,730,045.91) | Used (15 from $35.54)

Show ⊙New ○Prime offers only (0)

New 1-2 of 2 offers

Price + Shipping	Condition	Seller Information
$1,730,045.91 + $3.99 shipping	**New**	Seller: **profnath** Seller Rating: ★★★★★ **93% positive** over the past 12 m (8,193 total ratings) In Stock. Ships from NJ, United States. Domestic shipping rates and return policy. Brand new, Perfect condition, Satisfaction Guaranteed.
$2,198,177.95 + $3.99 shipping	**New**	Seller: **bordeebook** Seller Rating: ★★★★★ **93% positive** over the past 12 m (125,891 total ratings) In Stock. Ships from United States. Domestic shipping rates and return policy. New item in excellent condition. Not used. May be a publishe overstock or have slight shelf wear. Satisfaction guaranteed

COMPLEXITY

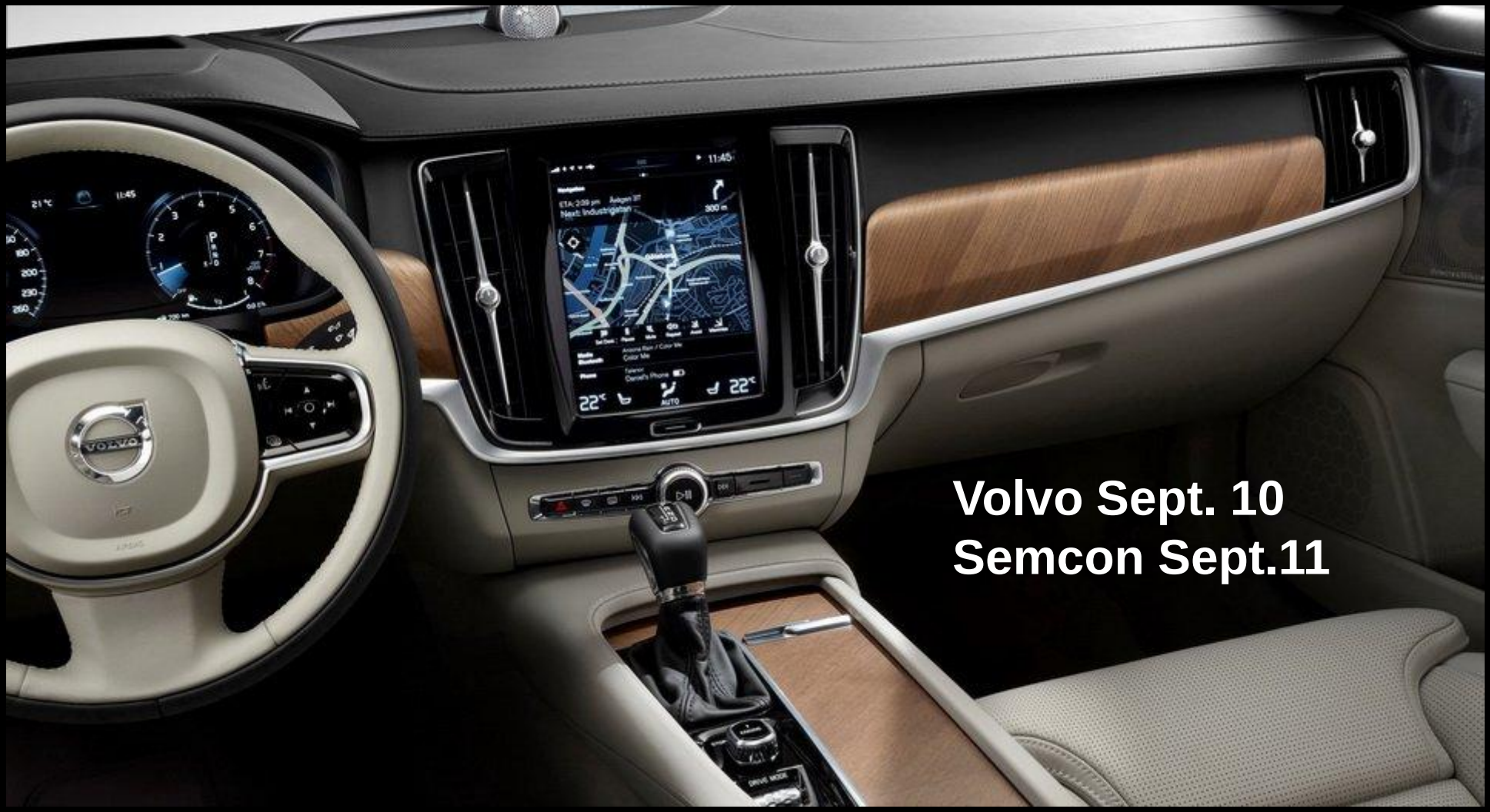
“The complexity of software is an essential property, not an accidental one.”

Fred Brooks, 1986

MLOC

- Simple app: 0.001
- Windows NT 3.1, 1993: 5
- Firefox: 10
- MS Office 2013: 40
- Facebook: 60
- Modern car: 100

Course Project



SOFTWARE CRISIS

Projects running over-budget.
Projects running over-time.
Software was very inefficient.
Software was of low quality.
Software often did not meet requirements.
Projects were unmanageable and code difficult to maintain.
Software was never delivered.

The first NATO Software Engineering Conference, Germany,1968

SOFTWARE ENGINEERING

Systematic & disciplined approach
to the development and maintenance
of software
to assure quality of processes and products

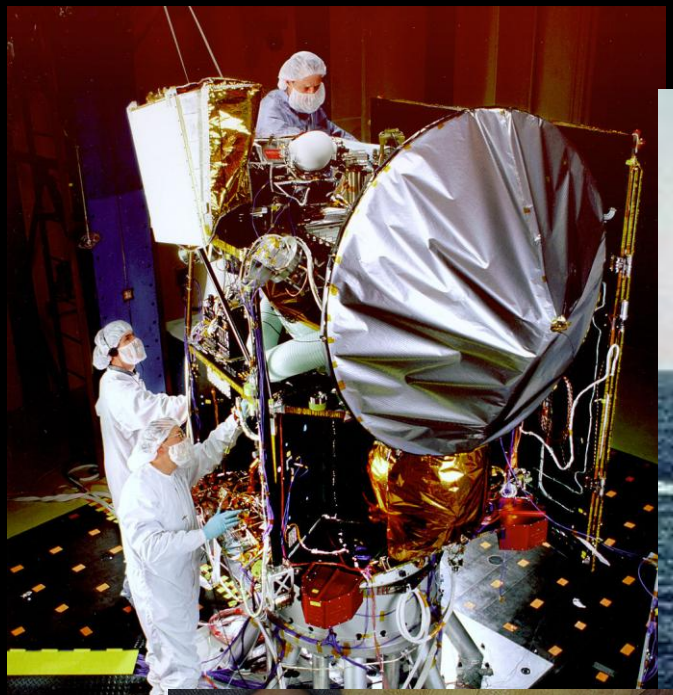
WATERFALL APPROACH

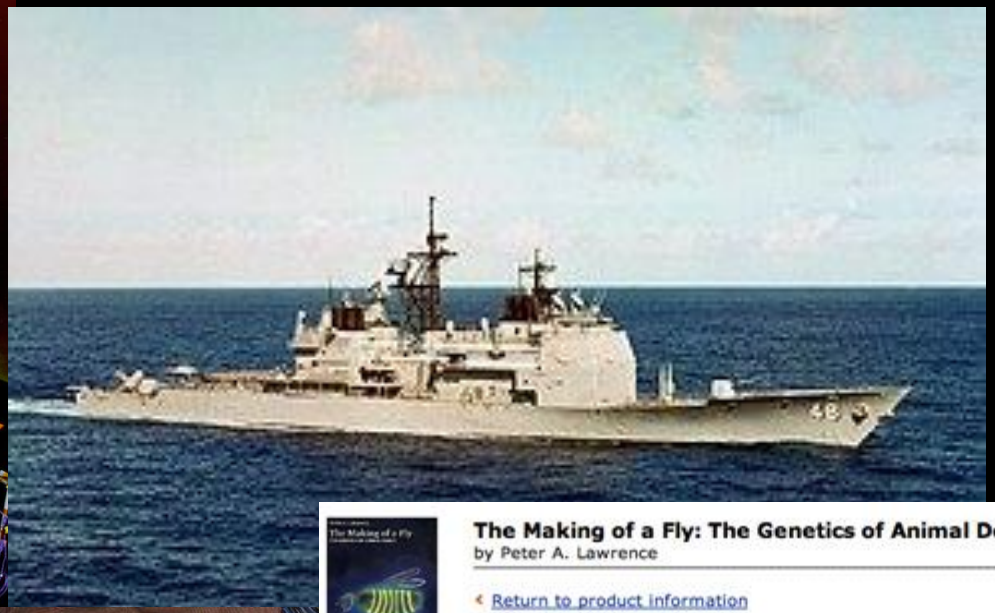
Specify
- Problem and solution
- Customer expectations

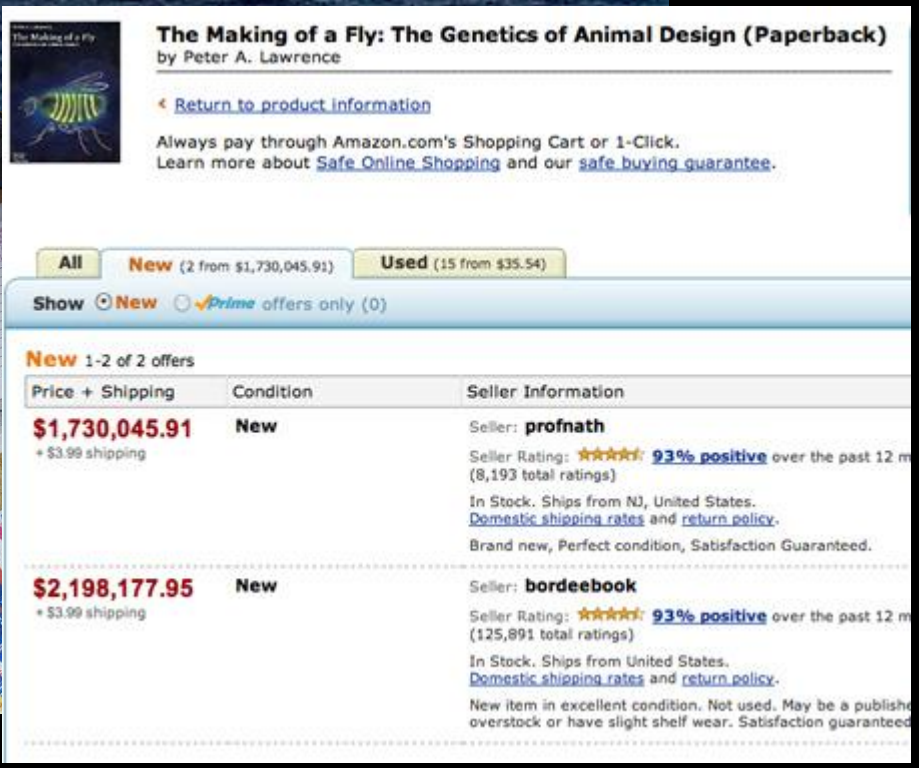
Implement
- Learn tools and technology
- Docs, configs, ...

Test
- Ensure quality

Evolve
- Debug
- Refine







The Making of a Fly: The Genetics of Animal Design (Paperback)
by Peter A. Lawrence

‹ Return to product information

Always pay through Amazon.com's Shopping Cart or 1-Click.
Learn more about Safe Online Shopping and our safe buying guarantee.

All | New (2 from $1,730,045.91) | Used (15 from $35.54)

Show ⊙New ○Prime offers only (0)

New 1-2 of 2 offers

Price + Shipping	Condition	Seller Information
$1,730,045.91 + $3.99 shipping	**New**	Seller: **profnath** Seller Rating: ★★★★★ **93% positive** over the past 12 m (8,193 total ratings) In Stock. Ships from NJ, United States. Domestic shipping rates and return policy. Brand new, Perfect condition, Satisfaction Guaranteed.
$2,198,177.95 + $3.99 shipping	**New**	Seller: **bordeebook** Seller Rating: ★★★★★ **93% positive** over the past 12 m (125,891 total ratings) In Stock. Ships from United States. Domestic shipping rates and return policy. New item in excellent condition. Not used. May be a publishe overstock or have slight shelf wear. Satisfaction guaranteed

COMPLEXITY

“The complexity of software is an essential property, not an accidental one.”

Continuous world → discrete system
Immateriality of software
Understanding problem domain
Managing development

AGILE MANIFEST

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.

AGILE

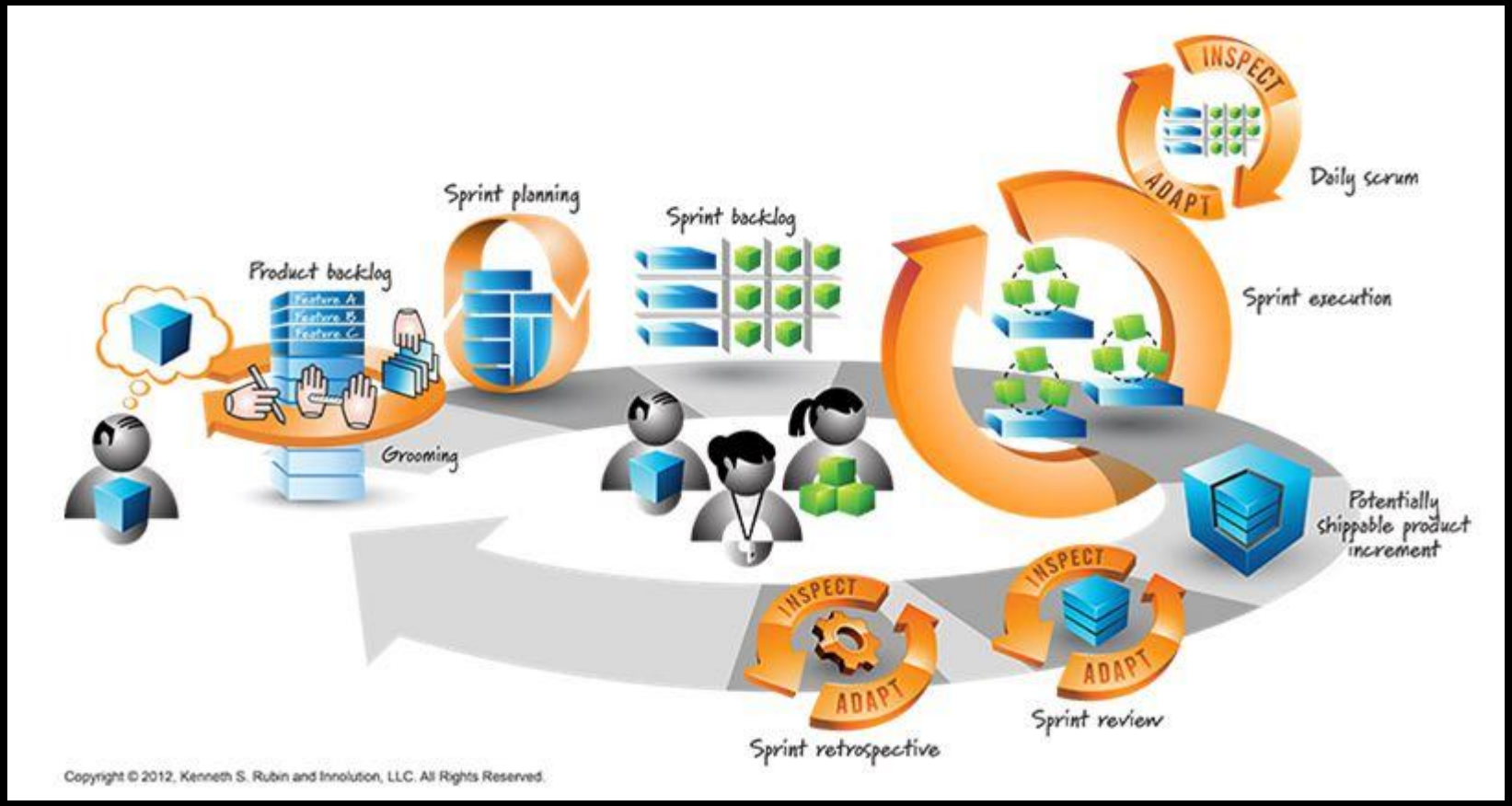
Processes

- Kanban
- XP - eXtreme Programming
- Test-driven development
- Feature-driven development
- Scrum

Practices

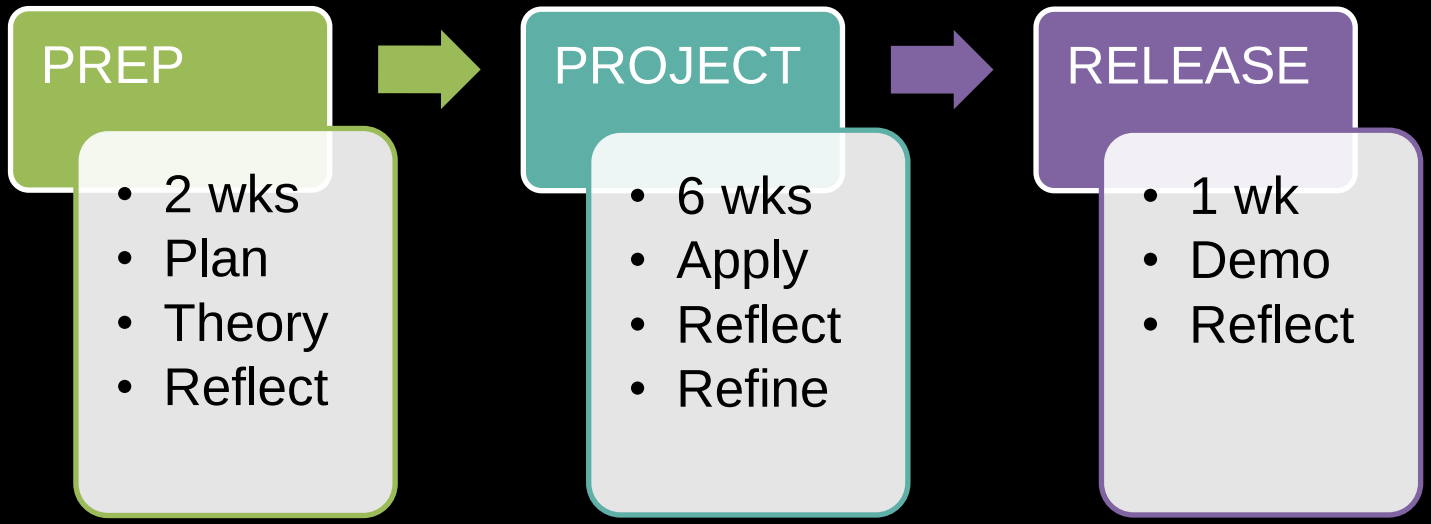
- User stories
- XFT Cross-functional Teams
- Stand-up meetings
- Short iterations
- Continuous testing
- Coding standards
- Sustainable pace
- Pair programming
- Customer value

SCRUM


INSPECT
ADAPT
Daily scrum
Sprint planning
Sprint backlog
Product backlog
Sprint execution
Grooming
Potentially shippable product increment
INSPECT
ADAPT
Sprint retrospective
INSPECT
ADAPT
Sprint review
Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.


Good
Morning

COURSE OVERVIEW




REFLECT?!

REFLECTION

What is
in relation to what might or should be and
includes feedback to reduce the gap

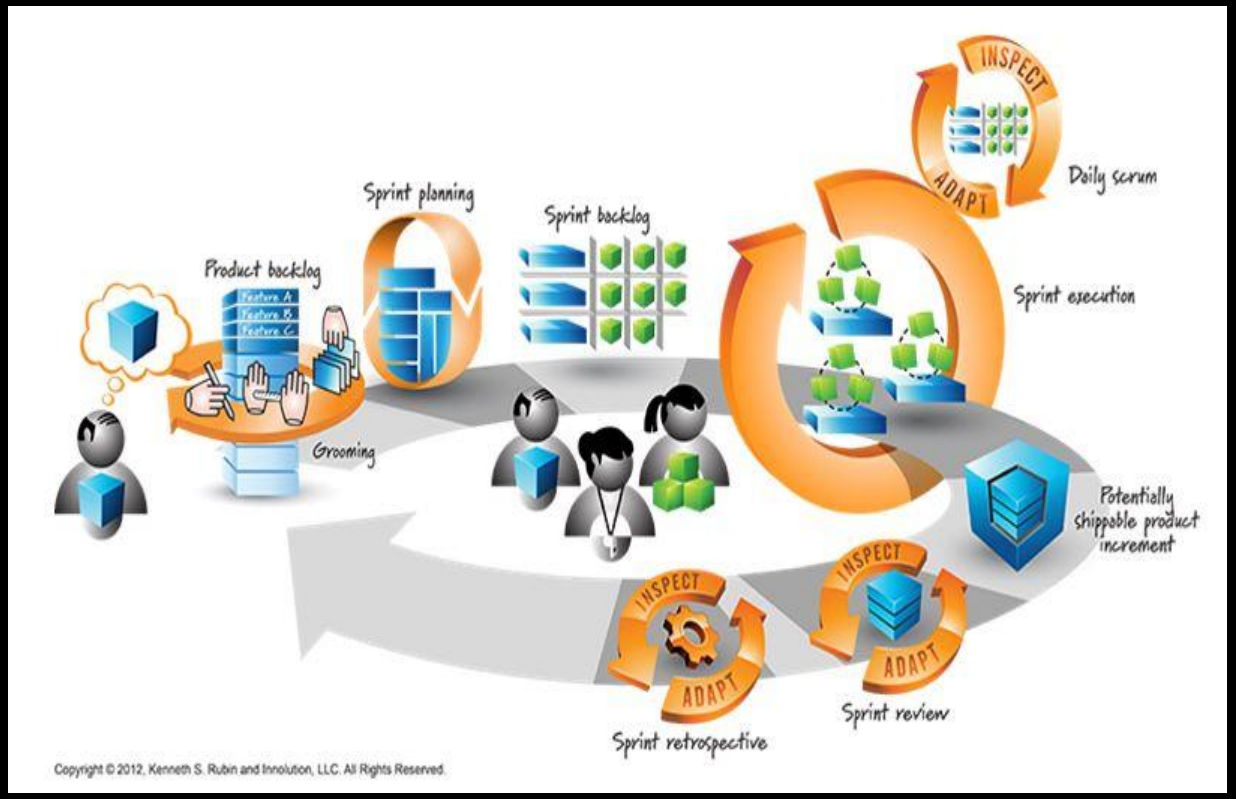
R. Smith. Formative Evaluation and the Scholarship of Teaching and Learning.
New Directions for Teaching and Learning, vol. 88, 2001, pp. 51-62

LEARNING GOALS

- Knowledge & Understanding
- Skills & Abilities
- Judgement & Approach

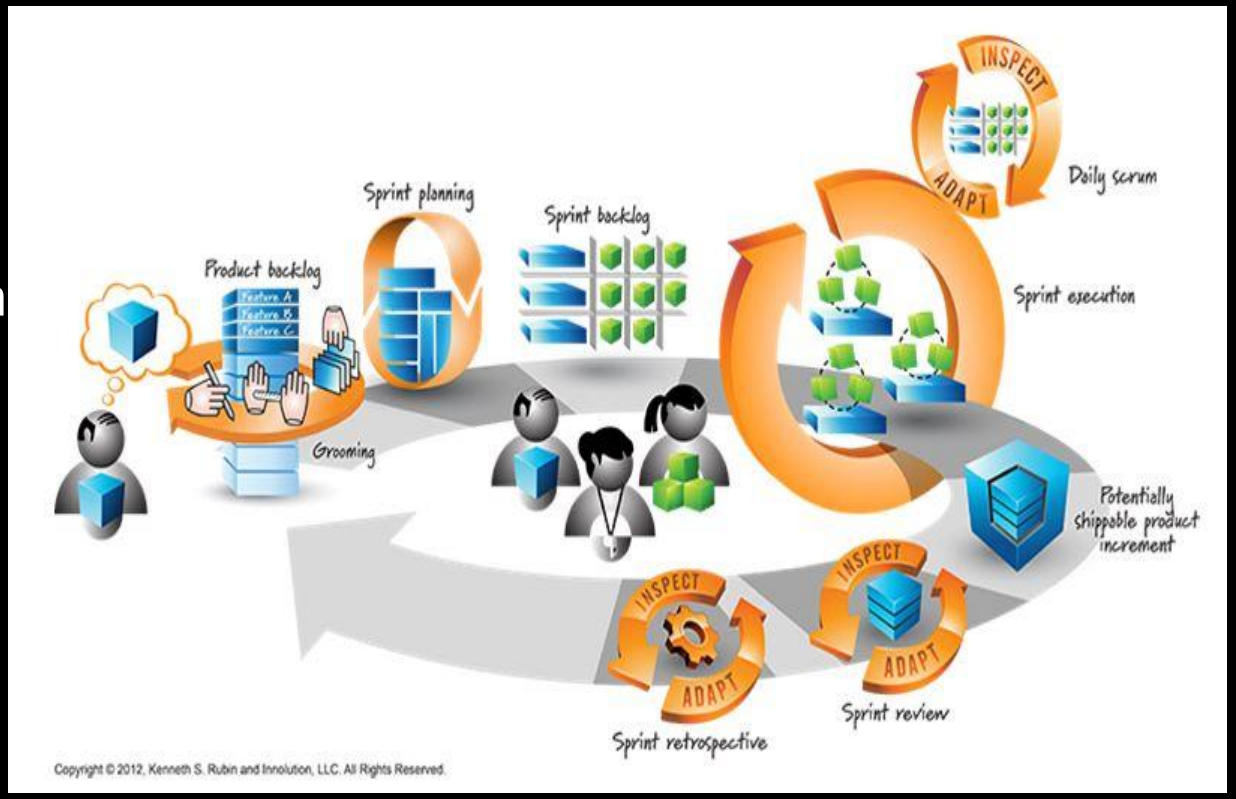
Knowledge and understanding, the student should be able to …

… describe software engineering as an engineering discipline by using relevant terminology
… describe the relationship between stakeholder, product, and process



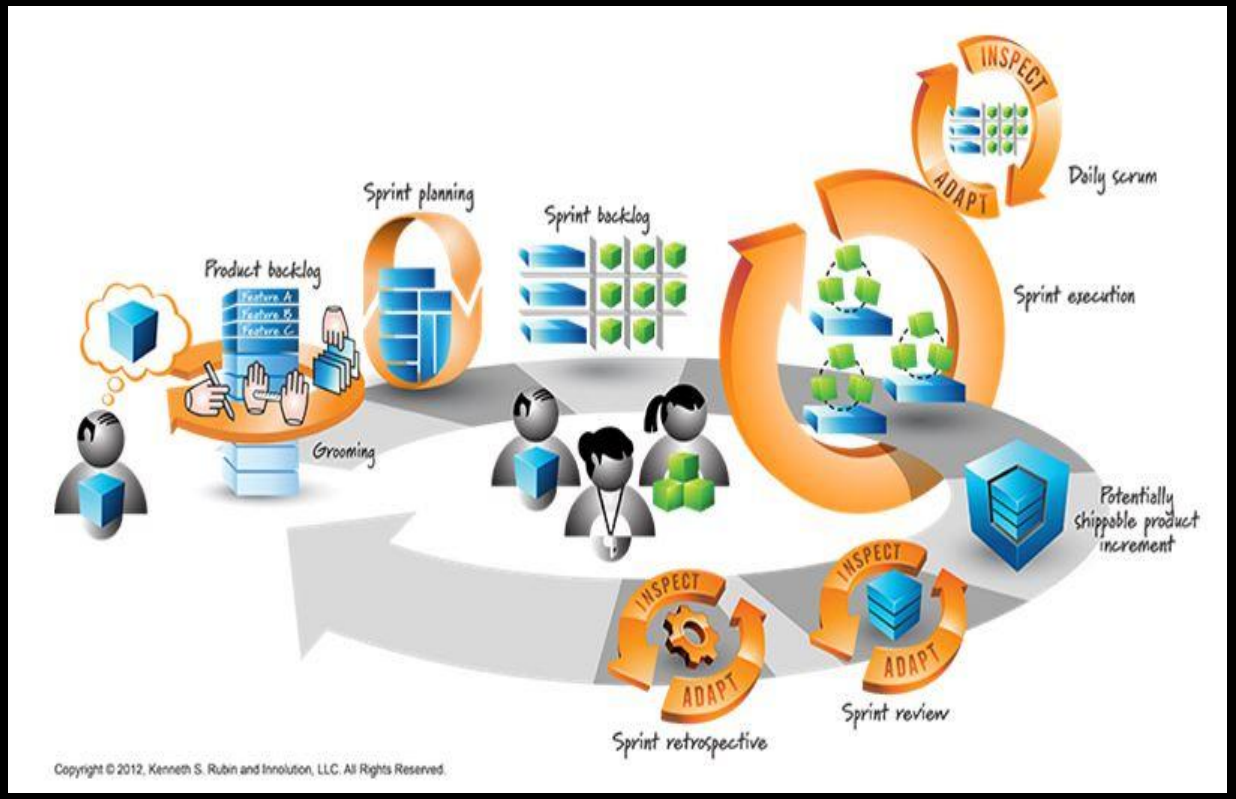
Skills and abilities, the student should be able to…

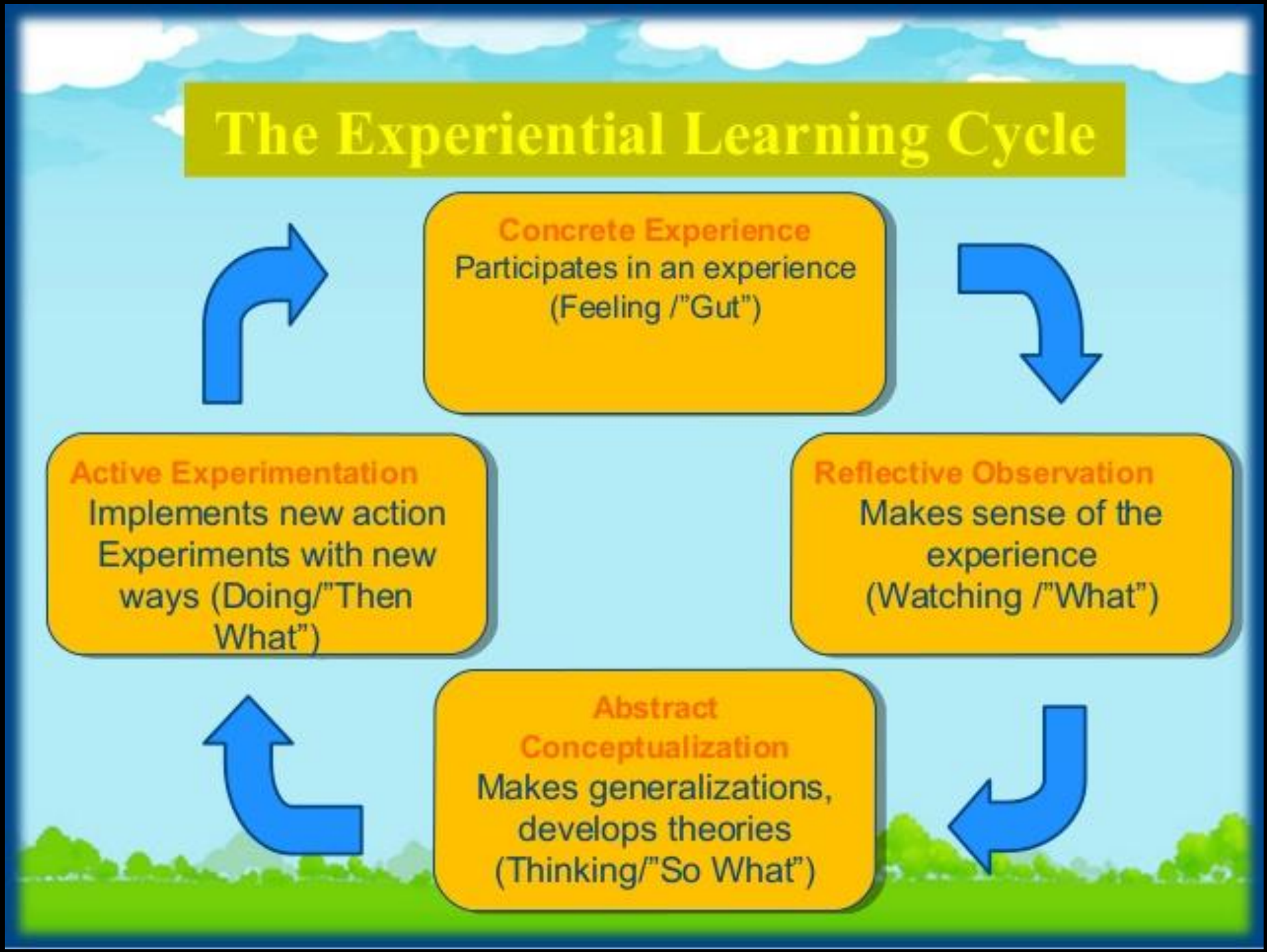
… specify, implement, and evaluate a system based on what different stakeholders perceive as valuable
… learn tools and APIs which are relevant for the project in collaboration with the other team members
… apply a structured software development process as a member of a team



Judgement and approach, the student should be able to…

… reflect on how the process was applied in a project
… reflect on the own and the team's learning strategies




The Experiential Learning Cycle
Concrete Experience
Participates in an experience
(Feeling /"Gut")
Reflective Observation
Makes sense of the
experience
(Watching /"What")
Abstract
Conceptualization
Makes generalizations,
develops theories
(Thinking/"So What")
Active Experimentation
Implements new action
Experiments with new
ways (Doing/"Then
What")

EACH WEEK

Reflect on 16 topics
Document reflection
Upload to team repo
Feedback during supervision
Final iteration is graded, 0-3p / topic

TEAM GRADES

Team report 0-48p
Individual reports 0-2p

U: 00 - 20p
3/G: 21 – 30p
4: 31 – 40p
5/VG: 41 - 50p

INDIVIDUAL GRADE

EACH WEEK:

what do I want to learn or understand better?

how can I help someone else, or the entire team, to learn something new?

what is my contribution towards the team’s application of scrum?

what is my contribution towards the team’s deliveries?

INDIVIDUAL GRADE

Upload documentation each week to team repo

One file with a heading for each week

Feedback during supervision

PERSONAL CONTRIBUTION

Individually

Total = size(Team) x 10
Score in range(0, Total)

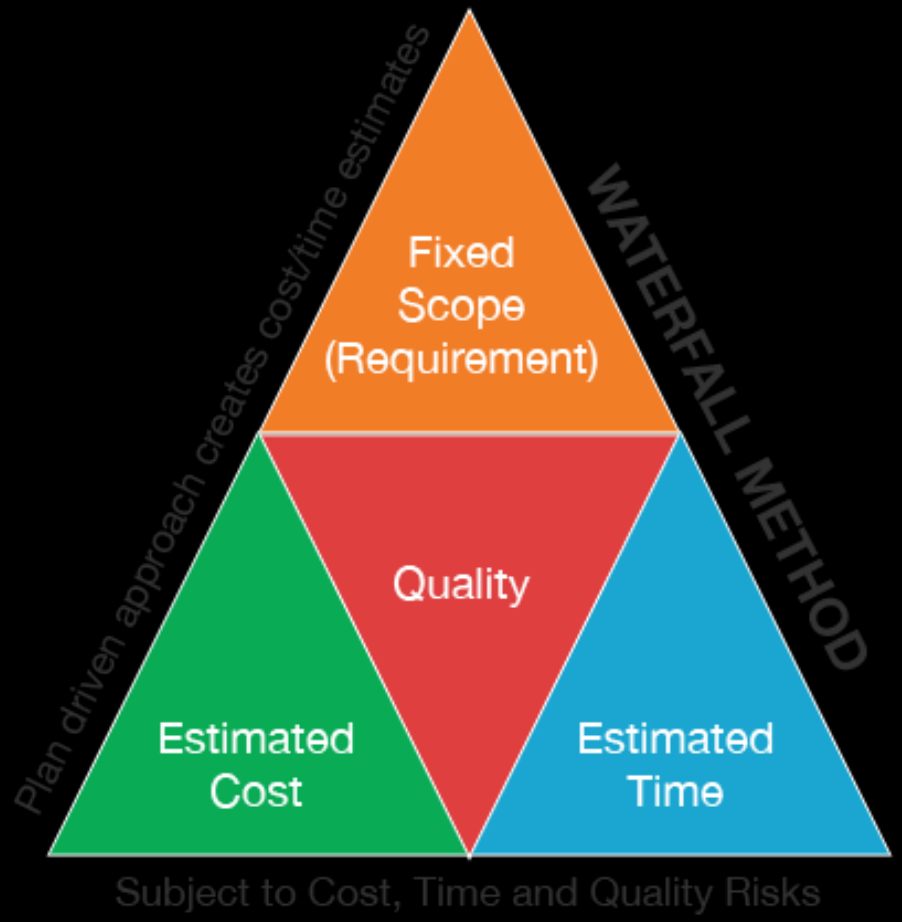
Link on the course homepage

	Eva	Per	Li	Jay	Foo	
Eva	12	5	11	14	8	50
Per	14	14	5	10	7	50
Li	13	12	5	10	10	50
Jay	14	12	5	14	7	50
Foo	15	10	5	13	7	50
	68	51	31	61	39	

INDIVIDUAL GRADE

Given all weekly assignments are completed:
Based on team grade
+/- for personal contribution
+/- gitinspector

Evidence for active contribution


Plan driven approach creates cost/time estimates
WATERFALL METHOD
Fixed
Scope
(Requirement)
Quality
Estimated
Cost
Estimated
Time
Subject to Cost, Time and Quality Risks

REALITY CHECK

What was the purpose of the lecture?
Which learning objectives were covered? How?
What was the relationship to the course assessment?

THIS WEEK

Monday: First exercise at 10:00 in Gamma
Tuesday: Scrum Lego exercise at 08:00 in Jupiter122
Wednesday: Lecture at 10:00 in Svea219

https://github.com/hburden/DAT256/

NEXT WEEK

Monday Volvo Cars at 10:00 in Svea226
Tuesday: Semcon toolchain at 10:00 in Svea219
Wednesday: Exercise at 10:00 in Gamma

https://github.com/hburden/DAT256

QA

'Questions don't have to make sense, Vincent', said Miss Susan.

'But answers do'

Terry Pratchett
Thief of Time, 2001