CS3213: Foundations of Software Engineering

In-class Lecture and Exercises

Bioblitz

- 147 observations
- 82 species
- 5 observers
- 53 identifiers

https://www.inaturalist.org/projects/nus-cs3213-2025

Mid-term Survey

Mid-term Feedback

Started: 8 Feb at 15:39

Quiz instructions

We have revised major parts of the course for this offering:

- We completely revised the project component and format
- We re-created the lab tutorial contents and slides to better ali
  contents and the course project
- We made the lecture contents more visual and went more in-depth compared to previous years
- We restructured the course to a blended-learning format with more in-person interactive activities

We would thus be grateful if you could spend some minutes to give us feedback on how these changes support your learning, hinder your learning, or any other feedback that you have that could help us to improve the class. We can try to address common feedback for the second part of the semester, or in next year's offering. Last year's feedback was already useful for us in identifying those components that required improvement, and we tried to address this feedback exhaustively.

The mid-term survey is anonymous.

Submit by the end of this week!

https://canvas.nus.edu.sg/courses/85437/quizzes/80001/take

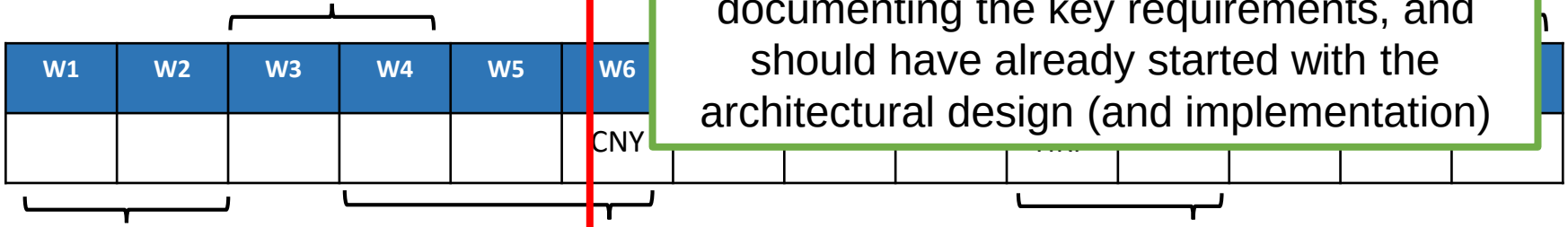
(Planned) Project Assignments

Assignment 2: Requirements Specification
Mon, 26 Jan to Mo, 10 Feb

Assignment 4: Intermediate Artifact
Mon,

Assignment 6: Final report and presentation
Apr

By now, you (mostly) finished eliciting and documenting the key requirements, and should have already started with the architectural design (and implementation)



Assignment 1: Requirements Elicitation Preparation
Mon, 12 Jan to Fri, 23 Jan

Assignment 3: Design Document
Mon, 2 Feb to Fri, 20 Feb

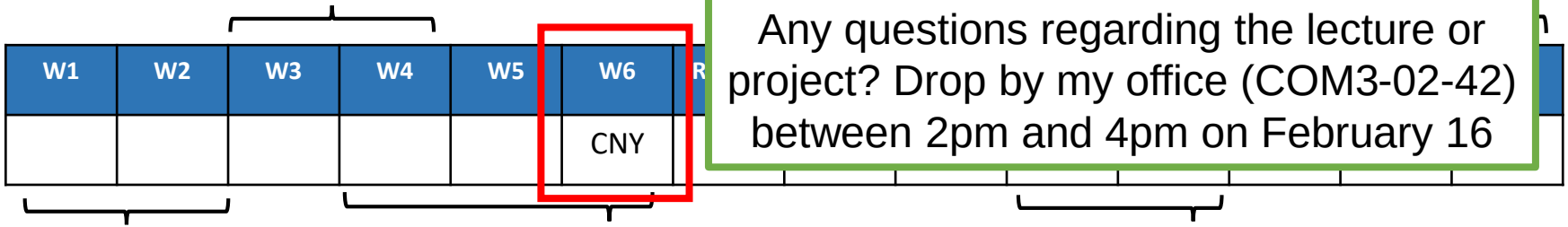
Assignment 5: Testing
Mon, 16 Mar to Fri, 27 Mar

(Planned) Project Assignments

Assignment 2: Requirements Specification
Mon, 26 Jan to Mo, 10 Feb

Assignment 4: Intermediate Artifact
Mon, 2 Mar to Fri, 13 Mar

Assignment 6: Final report and presentation
Mon, 30 Mar to Fri, 17 Apr



Any questions regarding the lecture or project? Drop by my office (COM3-02-42) between 2pm and 4pm on February 16

Assignment 1: Requirements Elicitation Preparation
Mon, 12 Jan to Fri, 23 Jan

Assignment 3: Design Document
Mon, 2 Feb to Fri, 20 Feb

Assignment 5: Testing
Mon, 16 Mar to Fri, 27 Mar

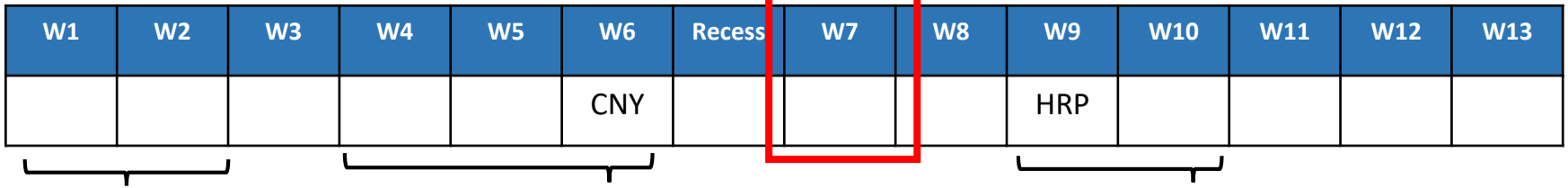
(Planned) Project Assignments

Assignment 2: Requirements Specification
Mon, 26 Jan to Mo, 10 Feb

Assignment 4: Intermediate Artifact
Mon, 2 Mar to Fri, 13 Mar

Assignment 6: Final report and presentation
Mon, 30 Mar to Fri, 17 Apr

Mid-term Exam!



Assignment 1: Requirements Elicitation Preparation
Mon, 12 Jan to Fri, 23 Jan

Assignment 3: Design Document
Mon, 2 Feb to Fri, 20 Feb

Assignment 5: Testing
Mon, 16 Mar to Fri, 27 Mar

Mid-term Exam (Main Venue)

- Date: Monday, March 2, 2026
- **Location: MPSH 1A**
- Hall Entry Time: 14:00
- **Start Time: 14:30**
- **End Time: 15:30**
- Hall Exit Time: 16:00
- Pen-and-paper and a double-sided A4 help sheet

Last Year's Mid-term Exam

CS3213 Mid-term Exam

6. (1 points) Which of these characteristics are typical of agile methods? Select all that apply.

   ◯ Trust in individuals and empowering teams
   ◯ Optimizing the development process
   ◯ Following a pre-defined plan
   ◯ Elaborate documentation

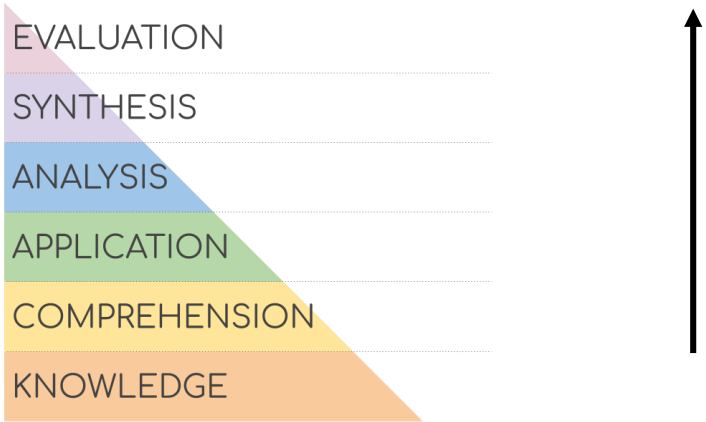
7. (1 points) Determine which of the following are true or false about *Extreme Programming (XP)*.

   Specifies concrete development practices such as Pair Programming and Collective Code Ownership. ◯ True ◯ False
   Various of its practices have become mainstream in both plan-driven as well as agile development. ◯ True ◯ False
   Extreme Programming was pioneered by the team working on Windows XP, hence the name. ◯ True ◯ False
   XP has parallels with DevOps, for example, the focus on Continuous Integration/Continuous Deployment (CI/CD). ◯ True ◯ False

Previous mid-term exams are online

This Year's Mid-term Exam

- More open-ended questions, few(er) multiple-choice questions
- Focus on the higher layers of Bloom's taxonomy
  - Fewer comprehension/knowledge questions



https://upload.wikimedia.org/wikipedia/commons/7/72/BloomsTaxonomy.png

FAQ

- In-class contents? Yes
- Project-specific? No
- Diagraming? Yes
- Questionnaire design? No
- Optional videos? No

(Planned) Project Assignments

Assignment 2: Requirements Specification
Mon, 26 Jan to Mo, 10 Feb

Assignment 4: Intermediate Artifact
Mon, 2 Mar to Fri, 13 Mar

Assignment 6: Final report and presentation
Mon, 30 Mar to Fri, 17 Apr

W1	W2	W3	W4	W5	W6	Recess	W7	W8	W9	W10	W11	W12	W13
					CNY				HRP				

Assignment 1:
Elicitation
Mon, 12 Jan

5: Testing
o Fri, 27 Mar

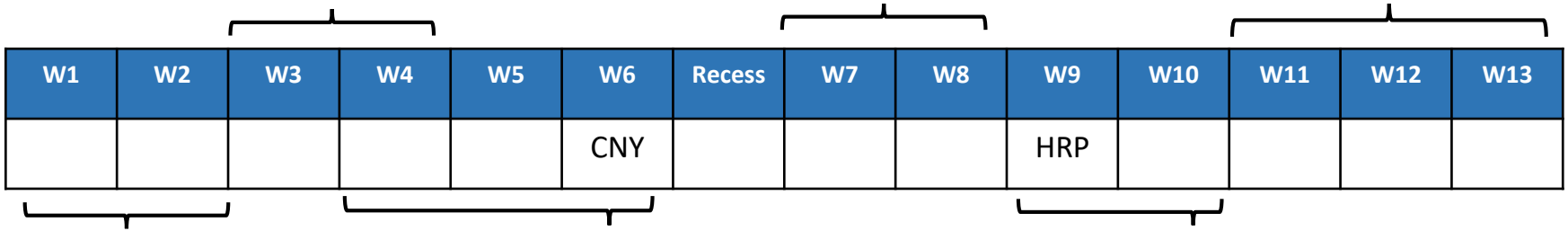
We strongly encourage you to work on the implementation **already now** (finish Assignment 3 asap!), since the submission deadline will be preceded by CNY, recess week, and the mid-term exam

(Planned) Project Assignments

Assignment 2: Requirements Specification
Mon, 26 Jan to Mo, 10 Feb

Assignment 4: Intermediate Artifact
Mon, 2 Mar to Fri, 13 Mar

Assignment 6: Final report and presentation
Mon, 30 Mar to Fri, 17 Apr



Assignment 1: Requirements Elicitation Preparation
Mon, 12 Jan to Fri, 23 Jan

Assignment 3: Design Document
Mon, 2 Feb to Fri, 20 Feb

Assignment 5: Testing
Mon, 16 Mar to Fri, 27 Mar

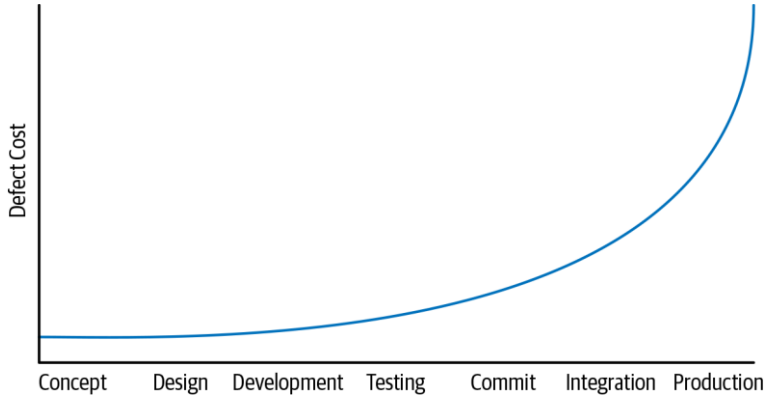
Week 1: Course Introduction

W1	W2	W3	W4	W5	W6	Recess	W7	W8	W9	W10	W11	W12	W13
L1	L2	L3	L4	L5				L6	L7	L8	L9	L10	

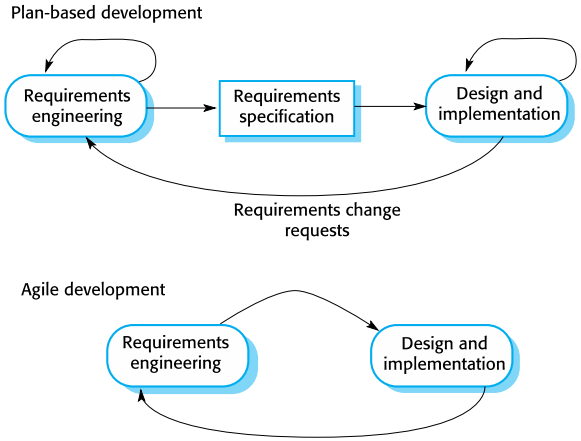
- Course overview
- What is software engineering?
  - Engineering discipline: messy real-world tradeoffs and decision making vs. elegant and clean principles
- *"No Silver Bullet"* essay

Week 1: Course Introduction

W1	W2	W3	W4	W5	W6	Recess	W7	W8	W9	W10	W11	W12	W13
L1	L2	L3	L4	L5				L6	L7	L8	L9	L10	



"Shifting Left"



"Plan-driven vs. agile development"

Week 2: Requirements Engineering (RE)

W1	W2	W3	W4	W5	W6	Recess	W7	W8	W9	W10	W11	W12	W13
L1	L2	L3	L4	L5				L6	L7	L8	L9	L10	

- Requirements elicitation and analysis
- Requirements specification
- Requirements validation
- Process models

Week 3: Project Overview and RE

W1	W2	W3	W4	W5	W6	Recess	W7	W8	W9	W10	W11	W12	W13
L1	L2	L3	L4	L5				L6	L7	L8	L9	L10	

Towards a Decade of Community Science Programme for Critically Endangered Primate

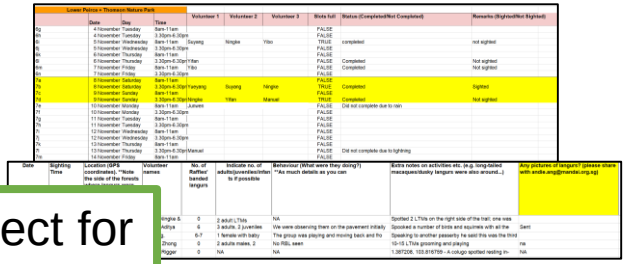

Primate Conservation 2021 (35): 179-188

Citizen Science Program for Critically Endangered Primates: A Case Study from Singapore

Andie Ang[1,2], Sabrina Jabbar[1], Vilma D'Rozario[3] and Jayasri Lakshminarayanan[1]

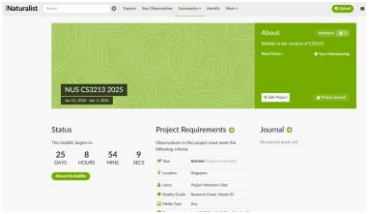
[1]*Raffles' Banded Langur Working Group, Wildlife Reserves Singapore Conservation Fund, Singapore*
[2]*Jane Goodall Institute, Singapore*
[3]*National Parks Board, Singapore*

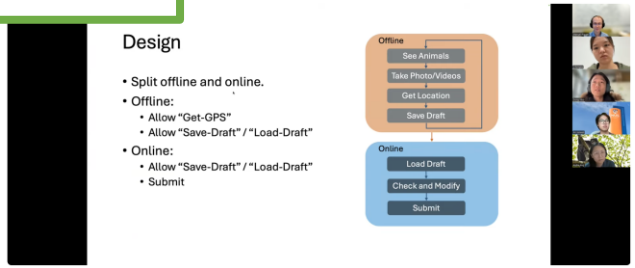
Abstract: Raffles' banded langur (*Presbytis femoralis*) is one of three species of non-human primates in Singapore. With only 67 individuals left, it is listed as locally Critically Endangered. Due to its elusive nature, arboreality, and small population



No questions about the project for the mid-term and final exam


Langur Survey with Wai Yee





Week 4: Modeling and Software Architecture

W1	W2	W3	W4	W5	W6	Recess	W7	W8	W9	W10	W11	W12	W13
L1	L2	L3	L4	L5				L6	L7	L8	L9	L10	

- UML and C4
- Attribute-driven Design (ADD)
- Architectural tactics and strategies

Week 5: (Agile) Software Frameworks

W1	W2	W3	W4	W5	W6	Recess	W7	W8	W9	W10	W11	W12	W13
L1	L2	L3	L4	L5				L6	L7	L8	L9	L10	

- Scrum
- Kanban
- Extreme Programming
- DevOps

Week 8: Software Testing

W1	W2	W3	W4	W5	W6	Recess	W7	W8	W9	W10	W11	W12	W13
L1	L2	L3	L4	L5				L6	L7	L8	L9	L10	

- Unit tests, integration tests, system tests
- Test-driven development
- Specification-based testing
- Structural testing

Week 9: Q/A with CHANG Sau Sheong

W1	W8	W9	W10	W11	W12	W13
L1	L6	L7	L8	L9	L10	

Shaping a digital nation: GovTech's Chang Sau Sheong on leading Singapore's tech evolution

Written by KrASIA Writers
Published on 5 Sep 2024 6 mins read

Share

Photo of Chang Sau Sheong, CTO and deputy chief executive (products) of GovTech.

GovTech's CTO discusses Singapore's digital trajectory, highlighting innovation, resilience, and the importance of events like the STACK Developer Conference.

https://kr-asia.com/shaping-a-digital-nation-govtechs-chang-sau-sheong-on-leading-singapores-tech-evolution

Week 10: Debugging

W1	W2	W3	W4	W5	W6	Recess	W7	W8	W9	W10	W11	W12	W13
L1	L2	L3	L4	L5				L6	L7	L8	L9	L10	

- “The Scientific Method” to Debugging
- Program slicing
- Statistical fault localization
- Test-case reduction
- Isolating failure-inducing changes

Week 11: Advanced Testing

W1	W2	W3	W4	W5	W6	Recess	W7	W8	W9	W10	W11	W12	W13
L1	L2	L3	L4	L5				L6	L7	L8	L9	L10	

- Property-based testing
- Differential testing
- Metamorphic testing
- Fuzzing

Week 12: Software Evolution and Course Summary

W1	W2	W3	W4	W5	W6	Recess	W7	W8	W9	W10	W11	W12	W13
L1	L2	L3	L4	L5				L6	L7	L8	L9	L10	

- Software versioning
- Hyrum’s Law
- Dependency management
- Deprecation

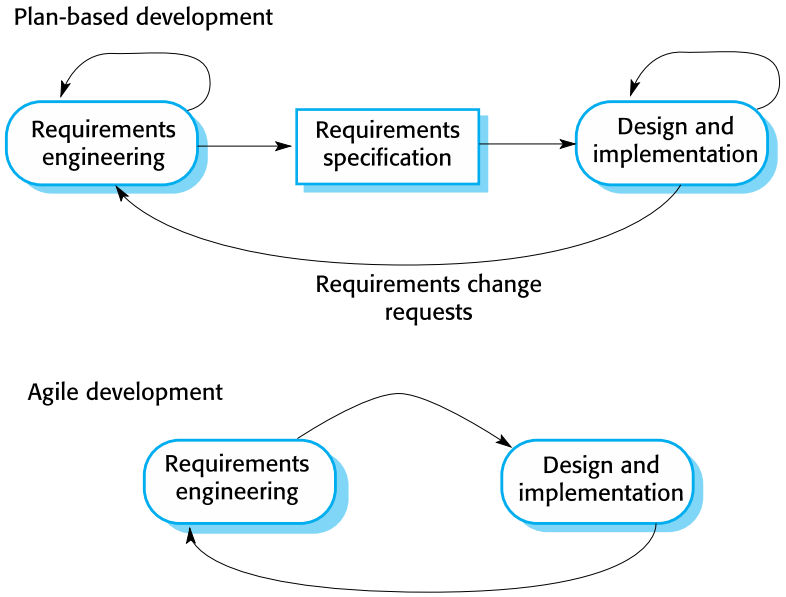
Week 13: Project Presentations

W1	W2	W3	W4	W5	W6	Recess	W7	W8	W9	W10	W11	W12	W13
L1	L2	L3	L4	L5				L6	L7	L8	L9	L10	

- Dr. Andie will join!

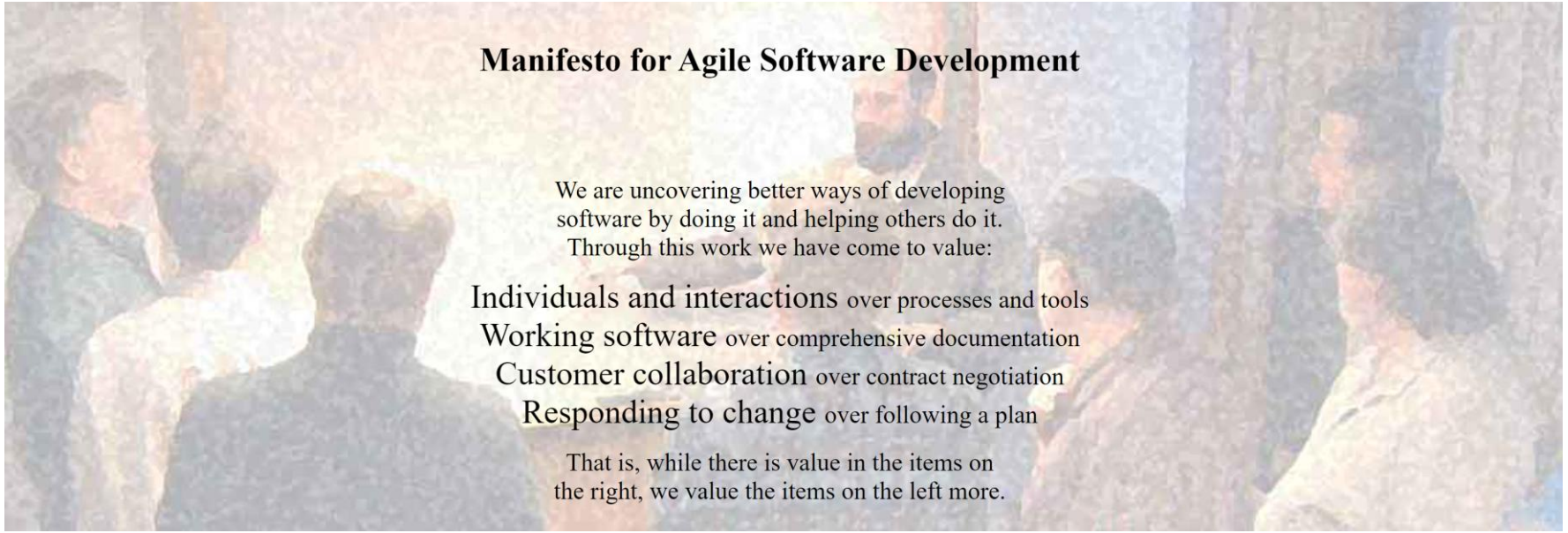
Recap

Plan-driven vs. Agile



Two general approaches to develop software: plan-driven and agile

Agile Manifesto (2001)



https://agilemanifesto.org/

The New New Product Development Game (1986)

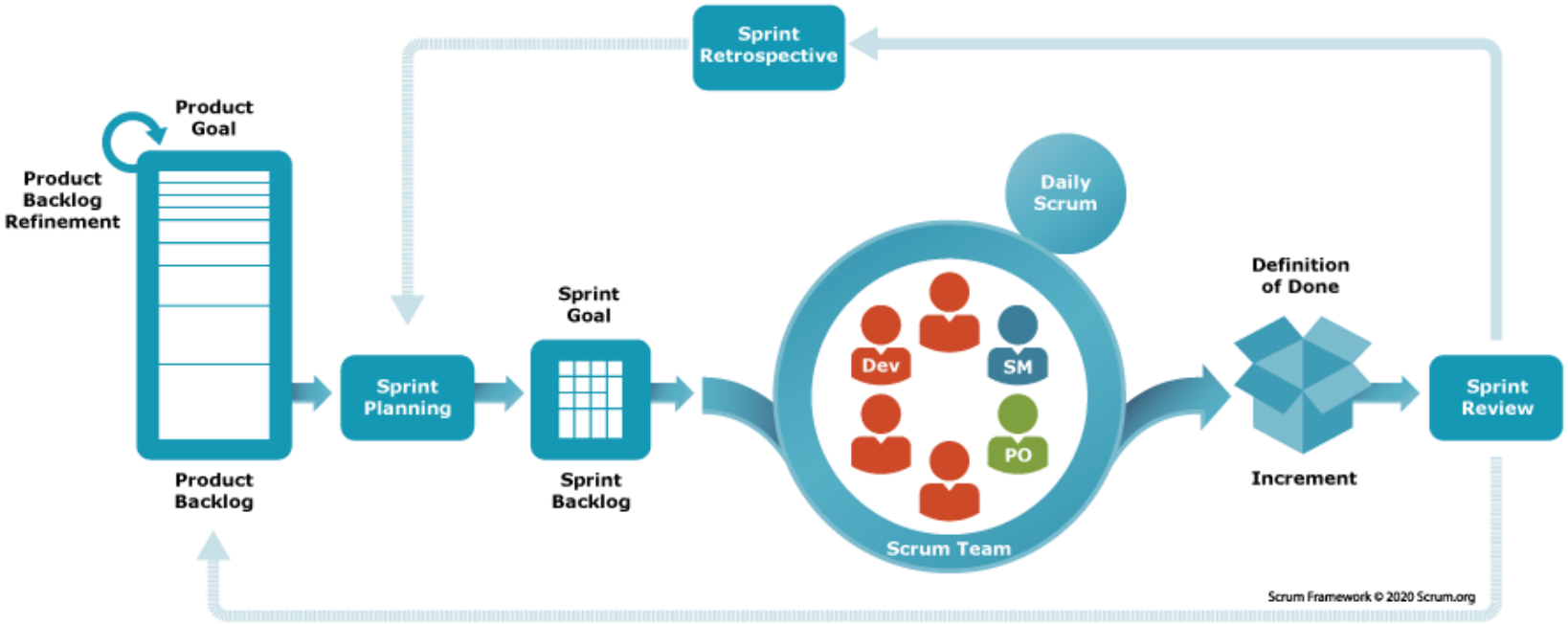
- New approach: fast and flexible, an “integrated” approach

“Scrum”

By PierreSelim - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=17336884

http://damiantgordon.com/Methodologies/Papers/The%20New%20Product%20Development%20Game.pdf

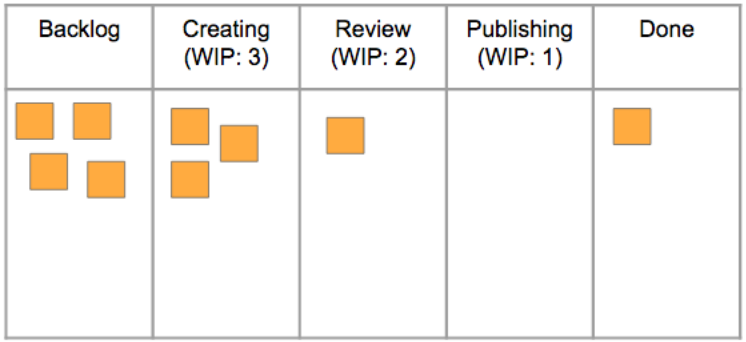
Scrum



https://www.scrum.org/resources/what-scrum-module

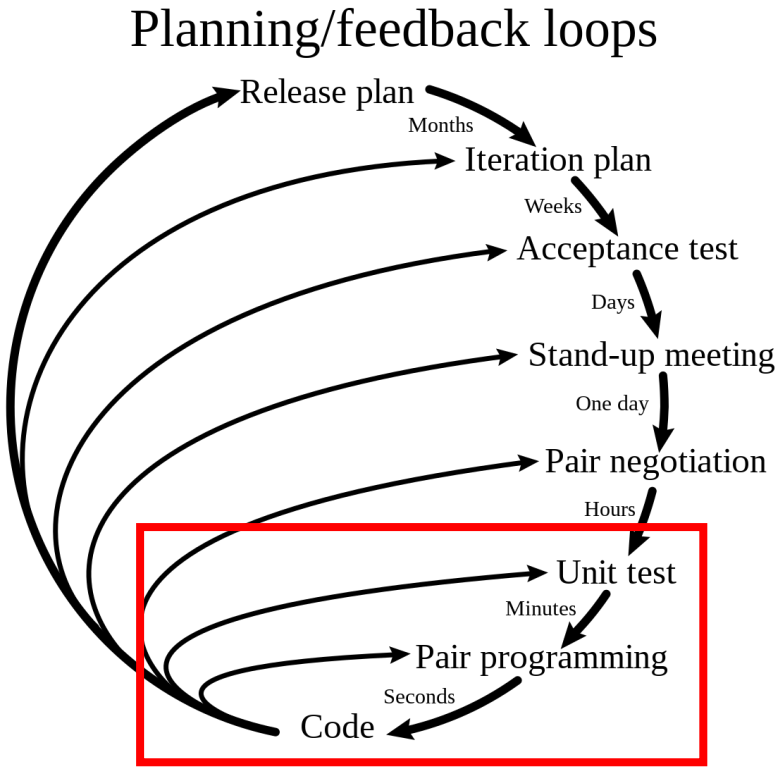
Flow

- ***Flow***: movement of potential value through a system
- Kanban optimizes *flow*
- Enforces a *work in progress (WIP) limit*



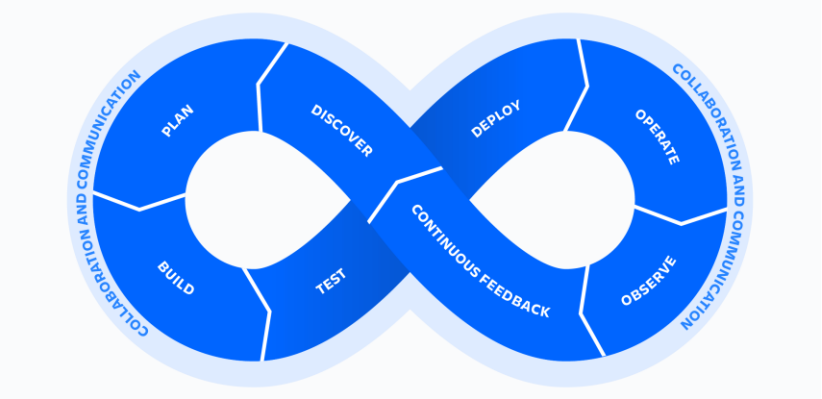
https://www.agilesherpas.com/blog/wip-limits-agile-marketing

Extreme Programming

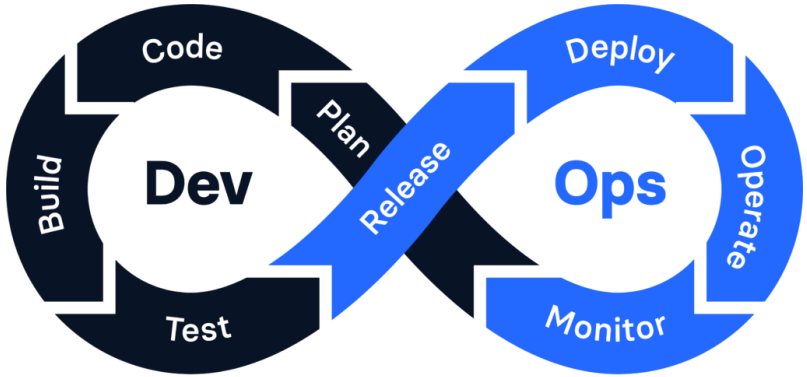


https://en.wikipedia.org/wiki/Extreme_programming

DevOps



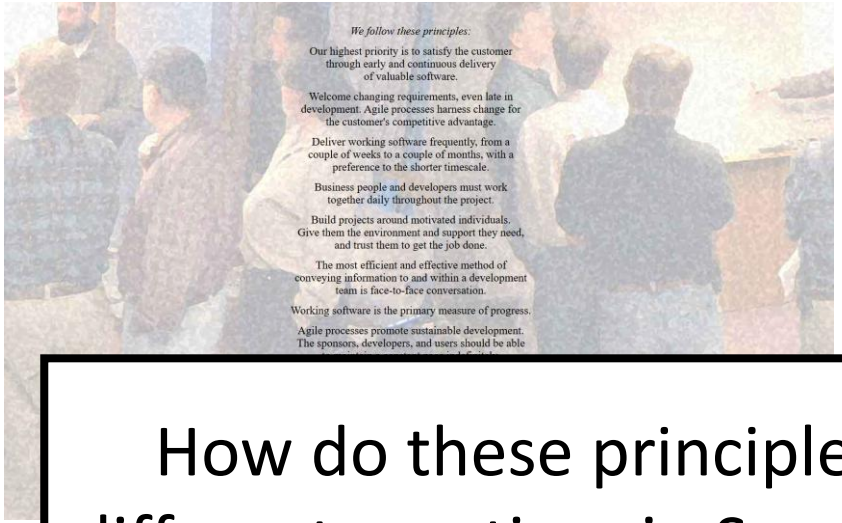
https://www.atlassian.com/devops



https://ubiqware.net/en/our-approach-to-devops/

Agile Exercises

Course Exercise



How do these principles map to the different practices in Scrum, Kanban, and DevOps?

https://pollev.com/manuelrigger

https://agilemanifesto.org/principles.html

Business people and developers must work together daily throughout the project.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Business people and developers must work together daily throughout the project.

- Scrum: Product Owner role
- Extreme Programming: on-site customer
- DevOps: shared responsibility and breaking down silos

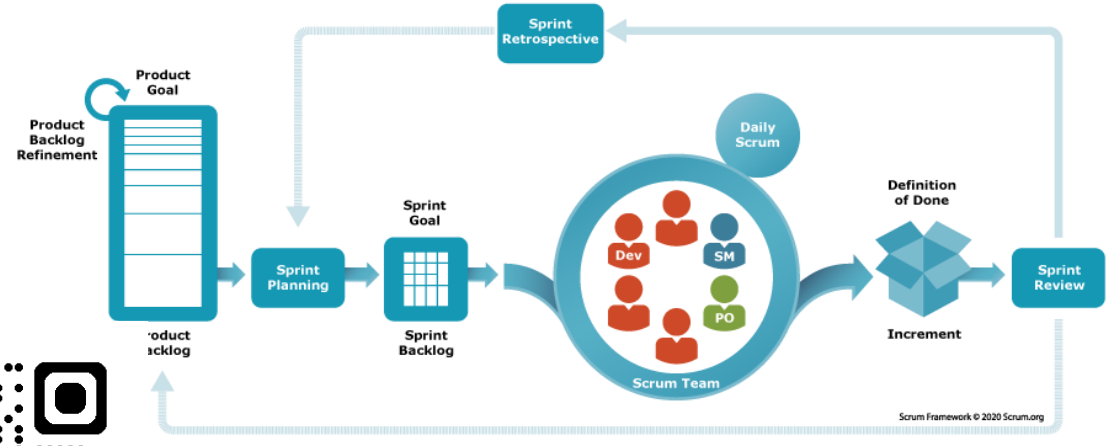
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

- Scrum: Sprints
- Kanban: Flow
- DevOps: continuous deployment, Value Stream Mapping

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

- Scrum: Sprint Retrospective
- Kanban: continuous improvement (Kaizen)
- DevOps: blameless postmortems, monitoring and continuous improvement

Course Exercise



Do you think that agile approaches are a silver bullet? Why/why not?

https://pollev.com/manuelrigger

Agile: Challenges with Adoption and Drawbacks

Have Agile Techniques been the Silver Bullet for Software Development at Microsoft?

Brendan Murphy
Microsoft Research
Cambridge, UK
bmurphy@microsoft.com

Christian Bird
Microsoft Research
Redmond, USA
cbird@microsoft.com

Thomas Zimmermann
Microsoft Research
Redmond, USA
tzimmer@microsoft.com

Laurie Williams
NCSU
Raleigh, USA
williams@csc.ncsu.edu

Nachiappan Nagappan
Microsoft Research
Redmond, USA
nachin@microsoft.com

Andrew Begel
Microsoft Research
Redmond, USA
andrew.begel@microsoft.com

***Abstract*—Background. The pressure to release high-quality, valuable software products at an increasingly faster rate is forcing software development organizations to adapt their development practices. Agile techniques began emerging in the mid-1990s in response to this pressure and to increased volatility of customer requirements and technical change. Theoretically, agile techniques seem to be the silver bullet for responding to these pressures on the software industry.**

Aims. This paper tracks the changing attitudes to agile adoption and techniques, within Microsoft, in one of the largest longitudinal surveys of its kind (2006-2012).

dology was used to develop these large software products. In reality, many large software companies did not religiously follow any specific development methodology and adapted methods and tools to suit the products they were producing.

Over time, consumers of software and software-intensive products increasingly welcomed more frequent software releases. Simultaneously, software began to be distributed electronically, and software-as-a-service (SaaS) increased in popularity. Traditional methodologies were viewed as too slow, not customer focused, not adaptable and too bureaucratic to handle the new software reality. In response, agile methods emerged in

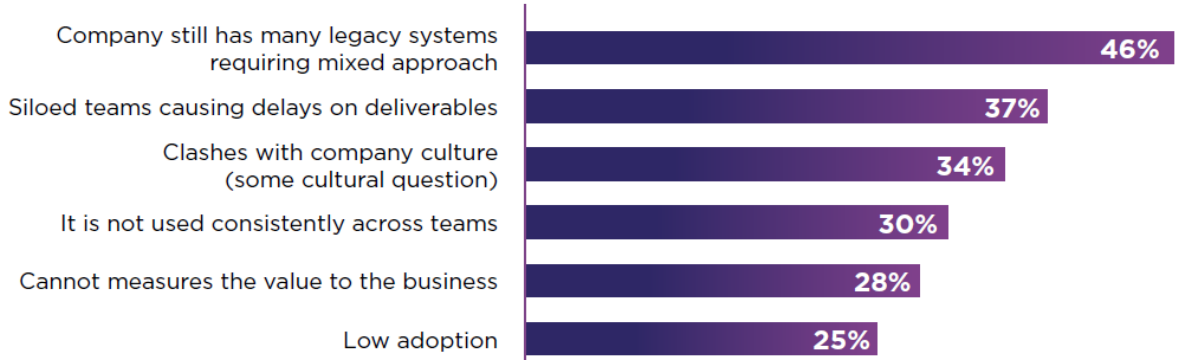
Agile: a Silver Bullet?

"[…] some proponents of agile (both at Microsoft and elsewhere) appear to become ***almost religious*** *about its use. These proponents* ***emphasize the potential benefits of using agile*** *while often* ***downplaying the cost or the learning curve****. We observed in some the attitude of "if agile doesn't work for you, then you're doing it wrong." Whether this is true or not (we do not claim either) is immaterial. In either case,* ***teams have attempted to adopt an agile practice and have run into problems****. Portraying agile as a nearly universal solution, downplaying its difficulties, or blaming the team when they do not reap the expected benefits, all serve to drive potential adopters away from agile practices."*

Agile: Problems with Adoption

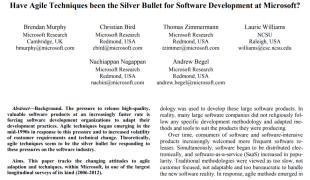
- Almost half of the survey takers perceive a general resistance to adopting agile practices
- Business teams do not understand what agile is and what it can do for them
- Only 6% of survey takers said they have no barriers to Agile adoption

Agile: Problem After Adoption



- “Too many mixed systems”, forcing practitioners to adopt hybrid approaches
- Siloed teams causing delays

Agile: a Silver Bullet?



Have Agile Techniques been the Silver Bullet for Software Development at Microsoft?

Brendan Murphy
Microsoft Research
Cambridge, UK
bmurphy@microsoft.com

Christian Bird
Microsoft Research
Redmond, USA
cbird@microsoft.com

Thomas Zimmermann
Microsoft Research
Redmond, USA
tzimmer@microsoft.com

Laurie Williams
NCSU
Raleigh, USA
williams@csc.ncsu.edu

Nachiappan Nagappan
Microsoft Research
Redmond, USA
nachin@microsoft.com

Andrew Begel
Microsoft Research
Redmond, USA
andrew.begel@microsoft.com

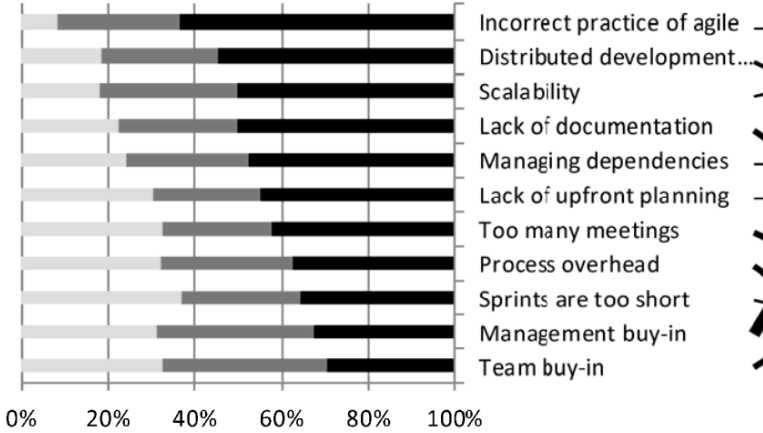
Abstract—Background. The pressure to release high-quality, valuable software products at an increasingly faster rate is forcing software development organizations to adapt their development practices. Agile techniques began emerging in the mid-1990s in response to this pressure and to increased volatility of customer requirements and technical change. Theoretically, agile techniques seem to be the silver bullet for responding to these pressures on the software industry.

Aim. This paper tracks the changing attitudes to agile adoption and techniques, within Microsoft, to one of the largest longitudinal surveys of its kind (2006-2012).

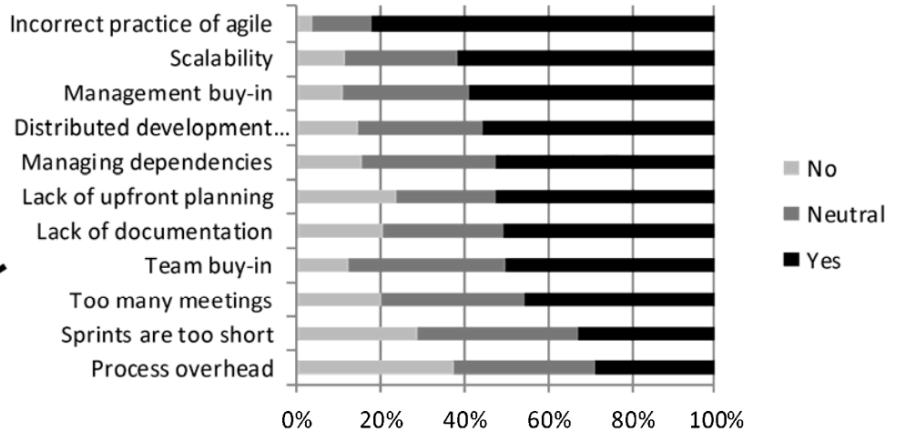
dology was used to develop these large software products. In reality, many large software companies did not religiously follow any specific development methodology and adapted methods and tools to suit the products they were producing.

Over time, consumers of software and software-intensive products increasingly welcomed more frequent software releases. Simultaneously, software began to be distributed electronically, and software-as-a-service (SaaS) increased in popularity. Traditional methodologies were viewed as too slow, not customer focused, not adaptable and too bureaucratic to handle the new software reality. In response, agile methods emerged in

Agile Devs: Perceived Problems

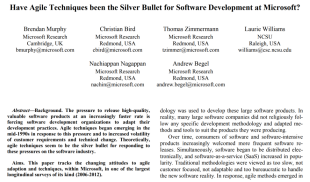


Non Agile Devs: Perceived Problems



https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Agile20Trends20ESEM20Master.pdf

Agile: a Silver Bullet?


Have Agile Techniques been the Silver Bullet for Software Development at Microsoft?

Brendan Murphy
Microsoft Research
Cambridge, UK
bmurphy@microsoft.com

Christian Bird
Microsoft Research
Redmond, USA
cbird@microsoft.com

Thomas Zimmermann
Microsoft Research
Redmond, USA
tzimmer@microsoft.com

Laurie Williams
NCSU
Raleigh, USA
williams@csc.ncsu.edu

Nachiappan Nagappan
Microsoft Research
Redmond, USA
nachin@microsoft.com

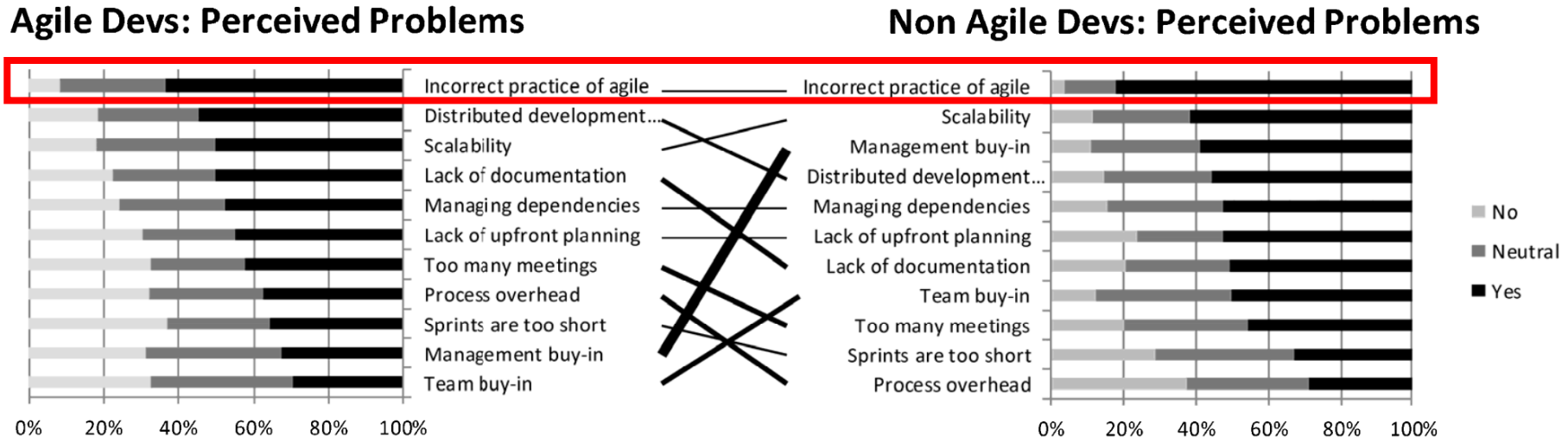
Andrew Begel
Microsoft Research
Redmond, USA
andrew.begel@microsoft.com

Abstract—Background. The pressure to release high-quality, valuable software products at an increasingly faster rate is forcing software development organizations to adapt their development practices. Agile techniques began emerging in the mid-1990s in response to this pressure and to increased volatility of customer requirements and technical change. Theoretically, agile techniques seem to be the silver bullet for responding to these pressures on the software industry.

Aims. This paper tracks the changing attitudes to agile adoption and techniques, within Microsoft, in one of the largest longitudinal surveys of its kind (2006-2012).

dology was used to develop these large software products. In reality, many large software companies did not religiously follow any specific development methodology and adapted methods and tools to suit the products they were producing.

Over time, consumers of software and software-intensive products increasingly welcomed more frequent software releases. Simultaneously, software began to be distributed electronically, and software-as-a-service (SaaS) increased in popularity. Traditional methodologies were viewed as too slow, not customer focused, not adaptable and too bureaucratic to handle the new software reality. In response, agile methods emerged in



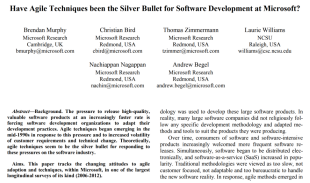
- Some organizations might claim to use an agile approach, but actually use a hybrid or traditional model

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Agile20Trends20ESEM20Master.pdf

Agile: a Silver Bullet?

https://medium.com/hackernoon/scrum-gone-wild-in-15-cartoons-cca23937a183

Agile: a Silver Bullet?


Have Agile Techniques been the Silver Bullet for Software Development at Microsoft?

Brendan Murphy
Microsoft Research
Cambridge, UK
bmurphy@microsoft.com

Christian Bird
Microsoft Research
Redmond, USA
cbird@microsoft.com

Thomas Zimmermann
Microsoft Research
Redmond, USA
tzimmer@microsoft.com

Laurie Williams
NCSU
Raleigh, USA
williams@csc.ncsu.edu

Nachiappan Nagappan
Microsoft Research
Redmond, USA
nachin@microsoft.com

Andrew Begel
Microsoft Research
Redmond, USA
andrew.begel@microsoft.com

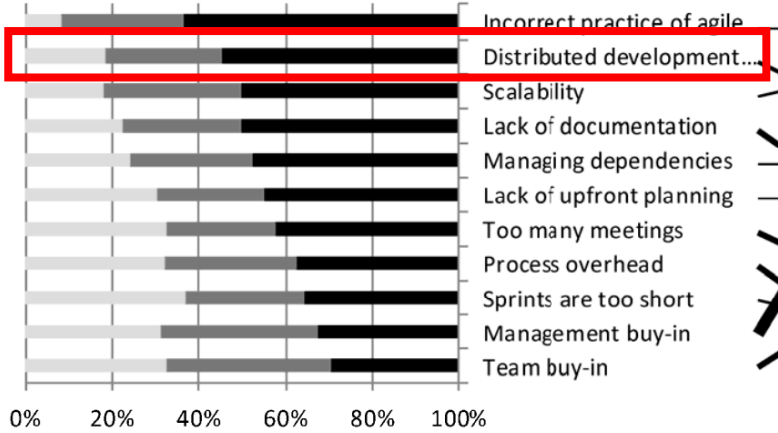
Abstract—Background. The pressure to release high-quality, valuable software products at an increasingly faster rate is forcing software development organizations to adapt their development practices. Agile techniques began emerging in the mid-1990s in response to this pressure and to increased volatility of customer requirements and technical change. Theoretically, agile techniques seem to be the silver bullet for responding to these pressures on the software industry.

Aims. This paper tracks the changing attitudes to agile adoption and techniques, within Microsoft, in one of the largest longitudinal surveys of its kind (2006-2012).

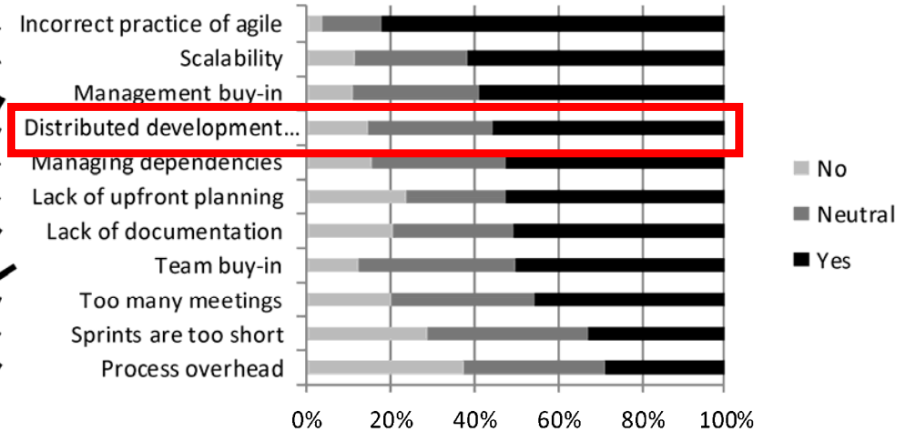
dology was used to develop these large software products. In reality, many large software companies did not religiously follow any specific development methodology and adapted methods and tools to suit the products they were producing.

Over time, consumers of software and software-intensive products increasingly welcomed more frequent software releases. Simultaneously, software began to be distributed electronically, and software-as-a-service (SaaS) increased in popularity. Traditional methodologies were viewed as too slow, not customer focused, not adaptable and too bureaucratic to handle the new software reality. In response, agile methods emerged in

Agile Devs: Perceived Problems



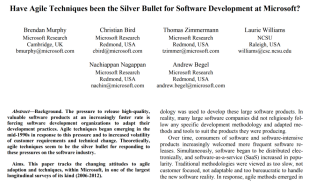
Non Agile Devs: Perceived Problems



- Microsoft has many geographically distributed teams

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Agile20Trends20ESEM20Master.pdf

Agile: a Silver Bullet?


Have Agile Techniques been the Silver Bullet for Software Development at Microsoft?

Brendan Murphy
Microsoft Research
Cambridge, UK
bmurphy@microsoft.com

Christian Bird
Microsoft Research
Redmond, USA
cbird@microsoft.com

Thomas Zimmermann
Microsoft Research
Redmond, USA
tzimmer@microsoft.com

Laurie Williams
NCSU
Raleigh, USA
williams@csc.ncsu.edu

Nachiappan Nagappan
Microsoft Research
Redmond, USA
nachin@microsoft.com

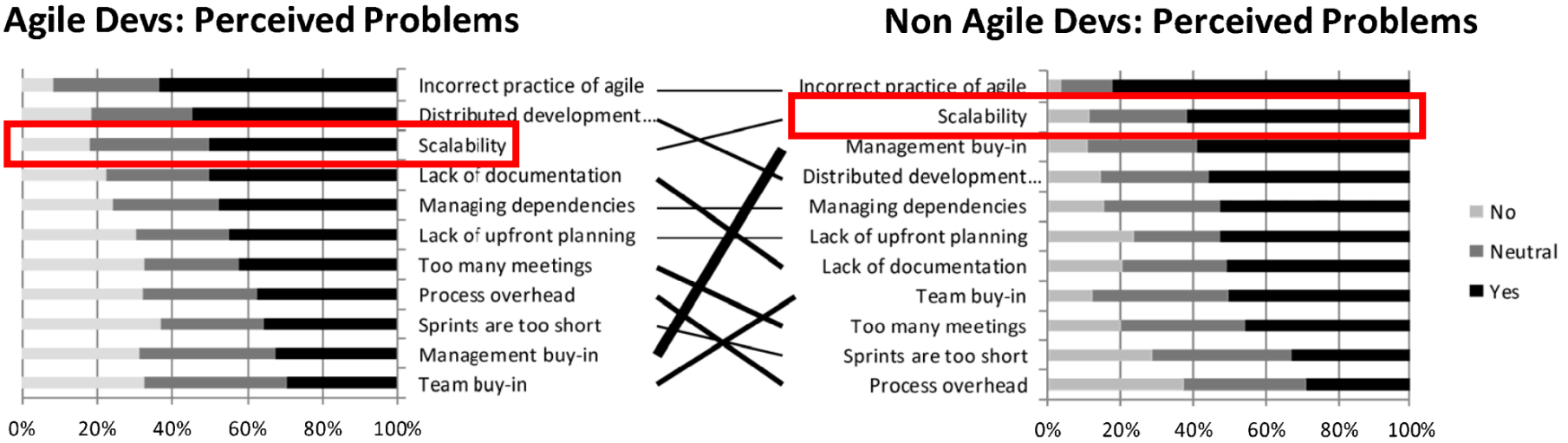
Andrew Begel
Microsoft Research
Redmond, USA
andrew.begel@microsoft.com

Abstract—Background. The pressure to release high-quality, valuable software products at an increasingly faster rate is forcing software development organizations to adapt their development practices. Agile techniques began emerging in the mid-1990s in response to this pressure and to increased volatility of customer requirements and technical change. Theoretically, agile techniques seem to be the silver bullet for responding to these pressures on the software industry.

Aims. This paper tracks the changing attitudes to agile adoption and techniques, within Microsoft, in one of the largest longitudinal surveys of its kind (2006-2012).

dology was used to develop these large software products. In reality, many large software companies did not religiously follow any specific development methodology and adapted methods and tools to suit the products they were producing.

Over time, consumers of software and software-intensive products increasingly welcomed more frequent software releases. Simultaneously, software began to be distributed electronically, and software-as-a-service (SaaS) increased in popularity. Traditional methodologies were viewed as too slow, not customer focused, not adaptable and too bureaucratic to handle the new software reality. In response, agile methods emerged in



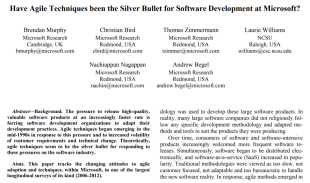
- Products like Microsoft Windows or Office have thousands of developers
- Multiple Product Owners, but causes some problems

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Agile20Trends20ESEM20Master.pdf

Scaling Scrum

- Various approaches have been proposed to scale Scrum
- Scaled Agile Framework (SAFe)
- Scrum@Scale (S@S)
- Nexus
- Large Scale Scrum (LeSS)
- Disciplined agile delivery (DAD)
- The “Spotify model”
- …

Agile: a Silver Bullet?



Have Agile Techniques been the Silver Bullet for Software Development at Microsoft?

Brendan Murphy
Microsoft Research
Cambridge, UK
bmurphy@microsoft.com

Christian Bird
Microsoft Research
Redmond, USA
cbird@microsoft.com

Thomas Zimmermann
Microsoft Research
Redmond, USA
tzimmer@microsoft.com

Laurie Williams
NCSU
Raleigh, USA
williams@csc.ncsu.edu

Nachiappan Nagappan
Microsoft Research
Redmond, USA
nachin@microsoft.com

Andrew Begel
Microsoft Research
Redmond, USA
andrew.begel@microsoft.com

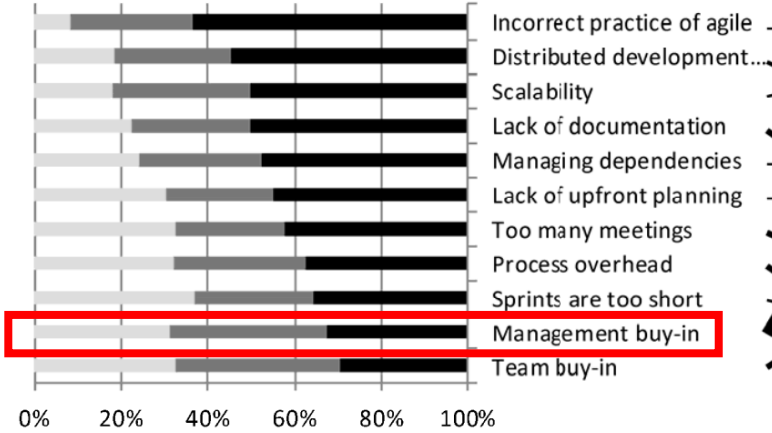
Abstract—Background. The pressure to release high-quality, valuable software products at an increasingly faster rate is forcing software development organizations to adapt their development practices. Agile techniques began emerging in the mid-1990s in response to this pressure and to increased volatility of customer requirements and technical change. Theoretically, agile techniques seem to be the silver bullet for responding to these pressures on the software industry.

Aim. This paper tracks the changing attitudes to agile adoption and techniques, within Microsoft, to one of the largest longitudinal surveys of its kind (2006-2012).

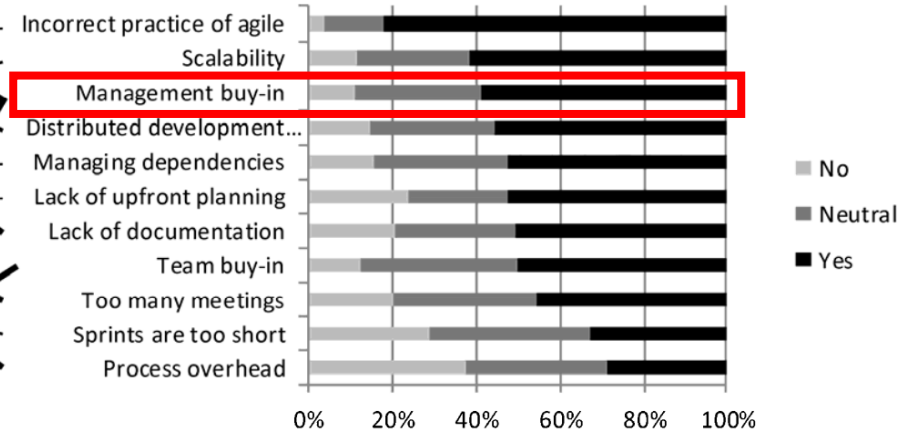
dology was used to develop these large software products. In reality, many large software companies did not religiously follow any specific development methodology and adapted methods and tools to suit the products they were producing.

Over time, consumers of software and software-intensive products increasingly welcomed more frequent software releases. Simultaneously, software began to be distributed electronically, and software-as-a-service (SaaS) increased in popularity. Traditional methodologies were viewed as too slow, not customer focused, not adaptable and too bureaucratic to handle the new software reality. In response, agile methods emerged in

Agile Devs: Perceived Problems



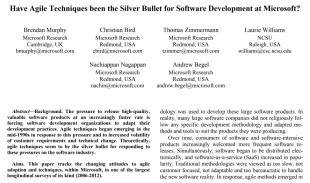
Non Agile Devs: Perceived Problems



- For agile practitioners, buy-in already exists
- Non-agile developers might have seen resistant to agile techniques

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Agile20Trends20ESEM20Master.pdf

Agile: a Silver Bullet?


Have Agile Techniques been the Silver Bullet for Software Development at Microsoft?

Brendan Murphy, Microsoft Research, Cambridge, UK, bmurphy@microsoft.com
Christian Bird, Microsoft Research, Redmond, USA, cbird@microsoft.com
Thomas Zimmermann, Microsoft Research, Redmond, USA, tzimmer@microsoft.com
Laurie Williams, NCSU, Raleigh, USA, williams@csc.ncsu.edu
Nachiappan Nagappan, Microsoft Research, Redmond, USA, nachin@microsoft.com
Andrew Begel, Microsoft Research, Redmond, USA, andrew.begel@microsoft.com

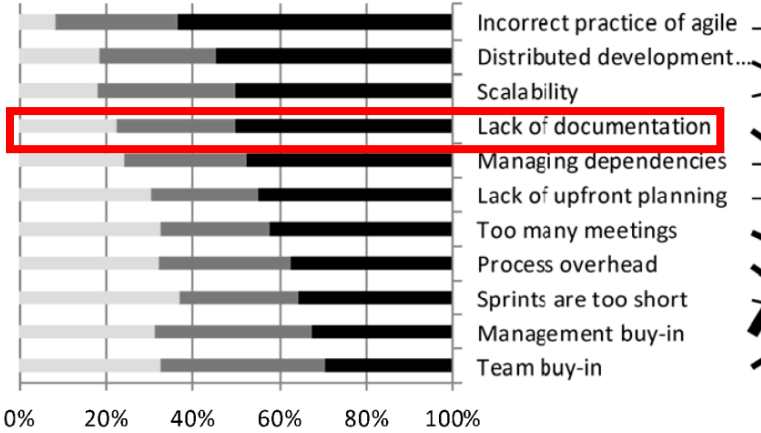
Abstract—Background. The pressure to release high-quality, valuable software products at an increasingly faster rate is forcing software development organizations to adopt their development practices. Agile techniques began emerging in the mid-1990s in response to this pressure and to increased volatility of customer requirements and technical change. Theoretically, agile techniques seem to be the silver bullet for responding to these pressures on the software industry.

Aims. This paper tracks the changing attitudes to agile adoption and techniques, within Microsoft, in one of the largest longitudinal surveys of its kind (2006-2012).

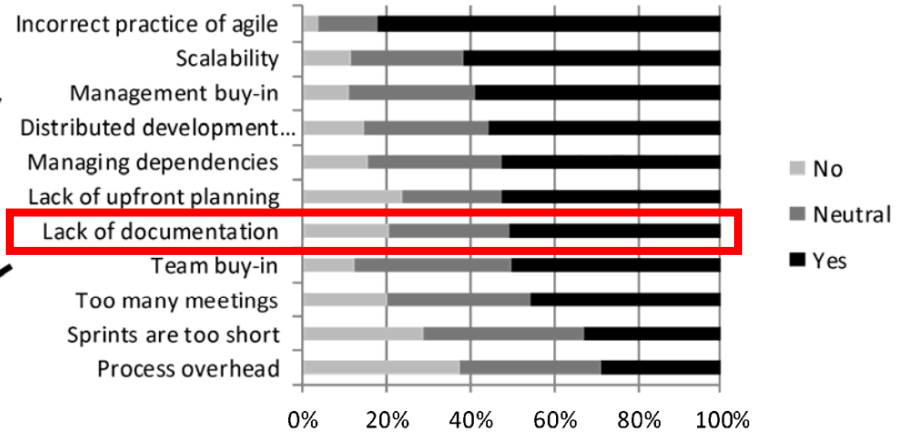
dology was used to develop these large software products. In reality, many large software companies did not religiously follow any specific development methodology and adapted methods and tools to suit the products they were producing.

Over time, consumers of software and software-intensive products increasingly welcomed more frequent software releases. Simultaneously, software began to be distributed electronically, and software-as-a-service (SaaS) increased in popularity. Traditional methodologies were viewed as too slow, not customer focused, not adaptable and too bureaucratic to handle the new software reality. In response, agile methods emerged in

Agile Devs: Perceived Problems



Non Agile Devs: Perceived Problems



- Agile developers might have “felt the pain” of not having documentation when they would have required it

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Agile20Trends20ESEM20Master.pdf

Planning in Agile

Numbers: 0, 1, 2, 3, 5, 8, 13

1
Changing the background color of the page to blue

3
Add an export button that downloads the data from database

5
Adding an admin dashboard

13
Train and integrate a ML-based classifier to distinguish different kinds of animals

Planning in Agile

Numbers: 0, 1, 2, 3, 5, 8, 13

???
Automatically resolve conflicts when multiple volunteers submit a report

1
Changing the background color of the page to blue

3
Add an export button that downloads the data from database

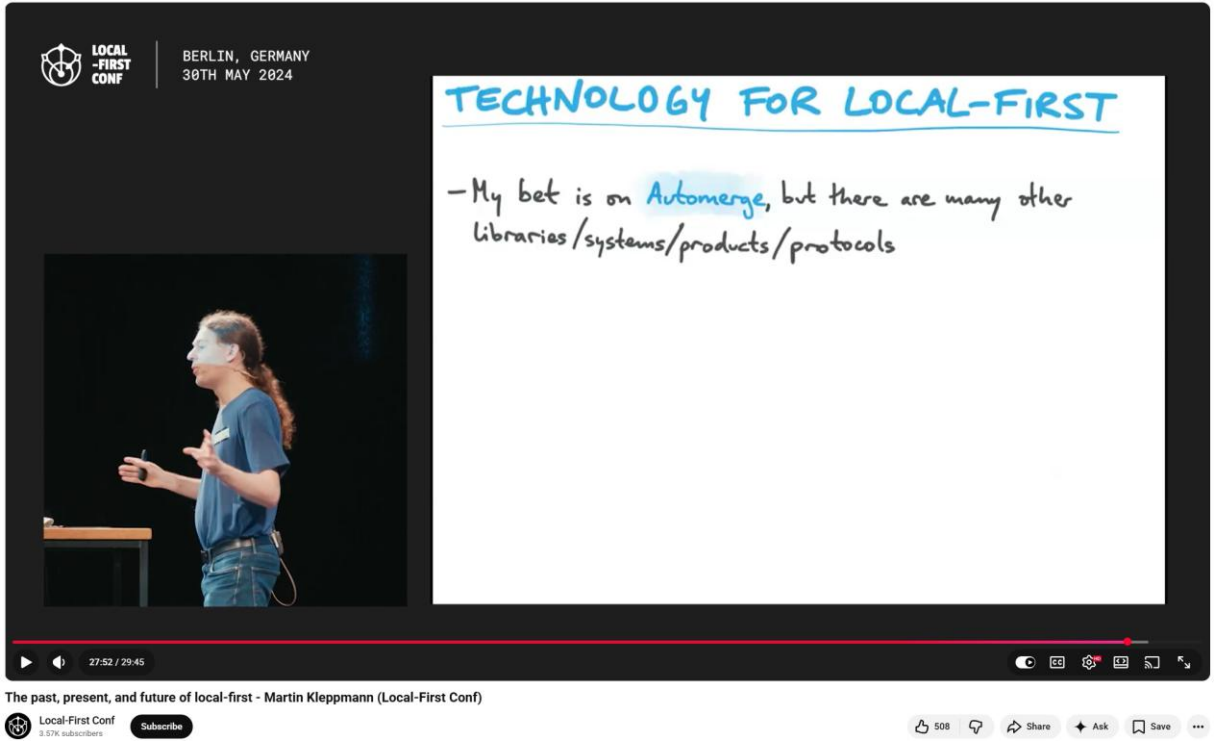
5
Adding an admin dashboard

13
Train and integrate a ML-based classifier to distinguish different kinds of animals

Planning Poker

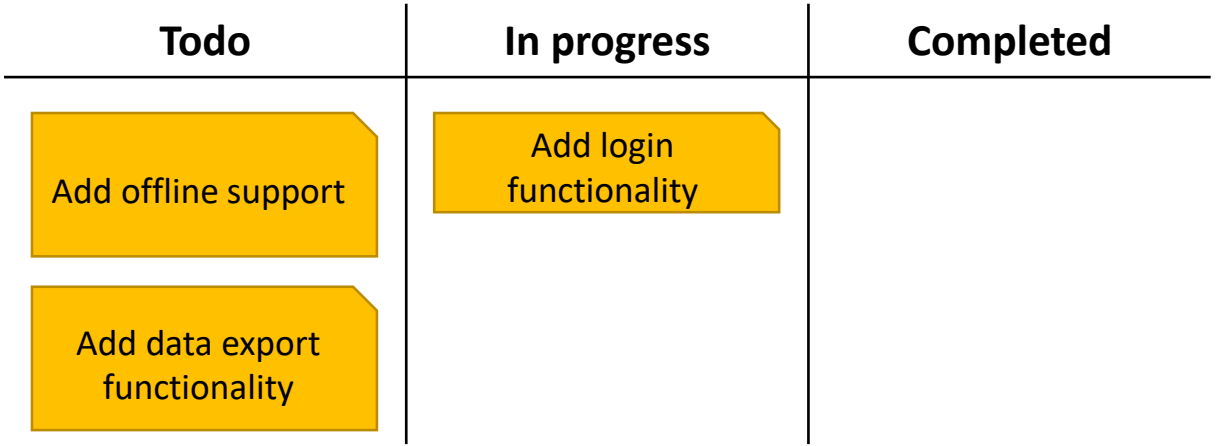
- Improves planning accuracy
- Estimating tasks relative to completed ones, rather than in absolute time
- Avoid bias and give equal voice to everyone

Local-First Software

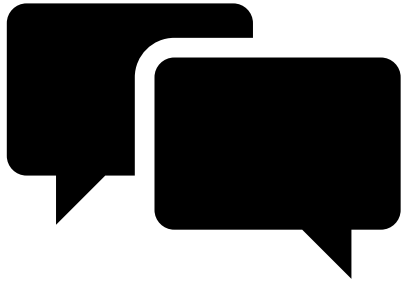


https://www.youtube.com/watch?v=NMq0vncHJvU

Kanban: Definition of Workflow



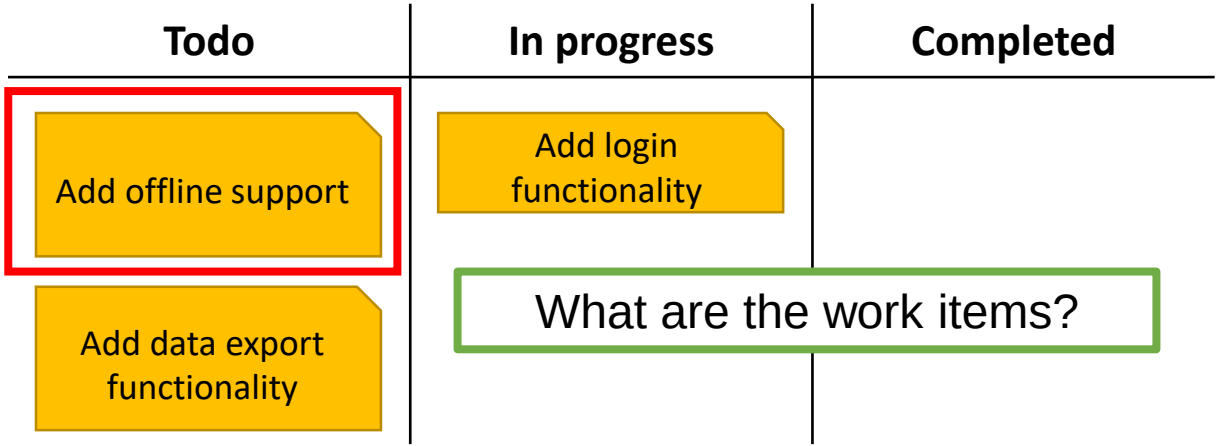
Definition of Workflow



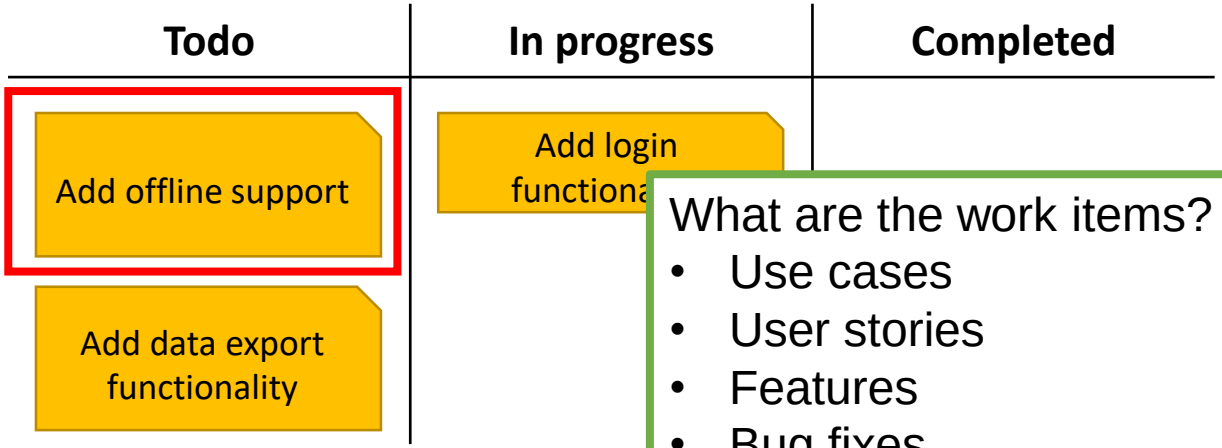
What concepts do we need to define for the Kanban *Definition of Workflow*?

https://pollev.com/manuelrigger

Definition of Workflow



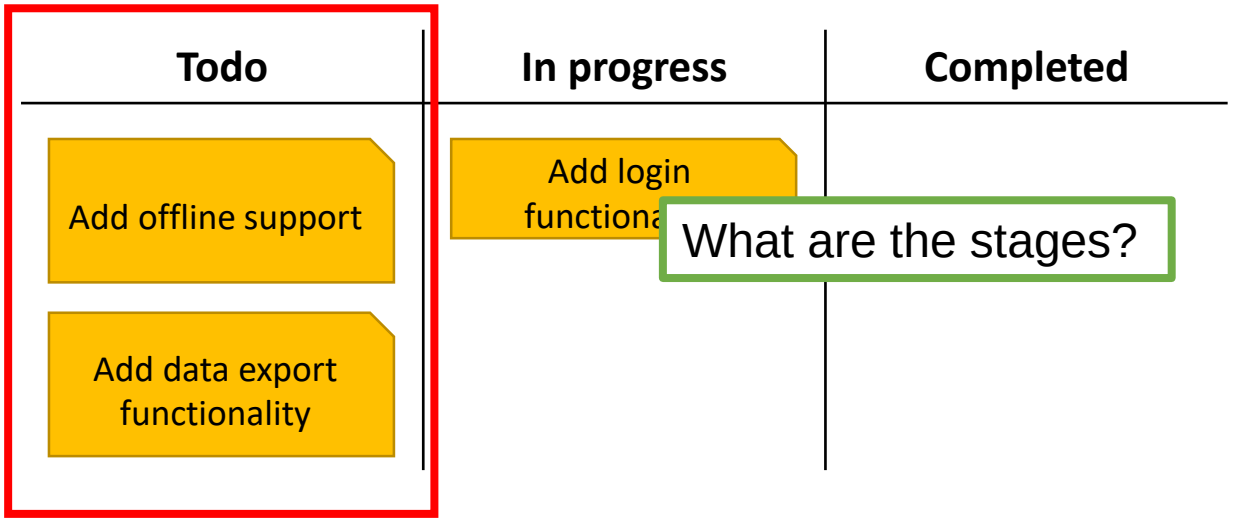
Definition of Workflow



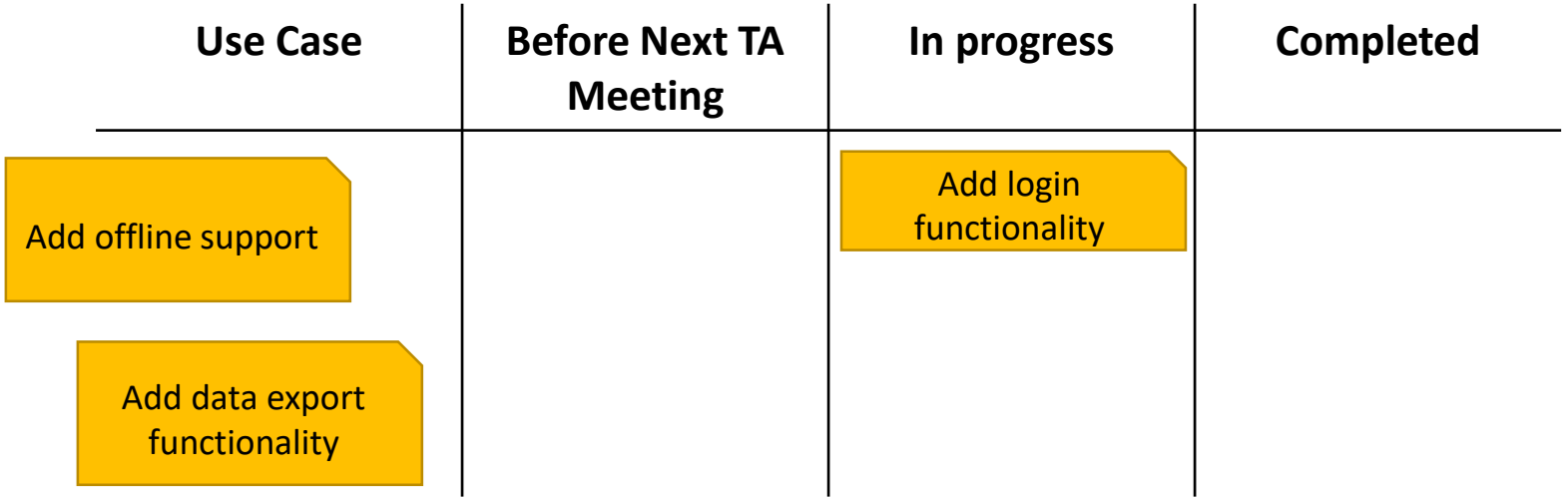
What are the work items?

- Use cases
- User stories
- Features
- Bug fixes
- Experiments
- …

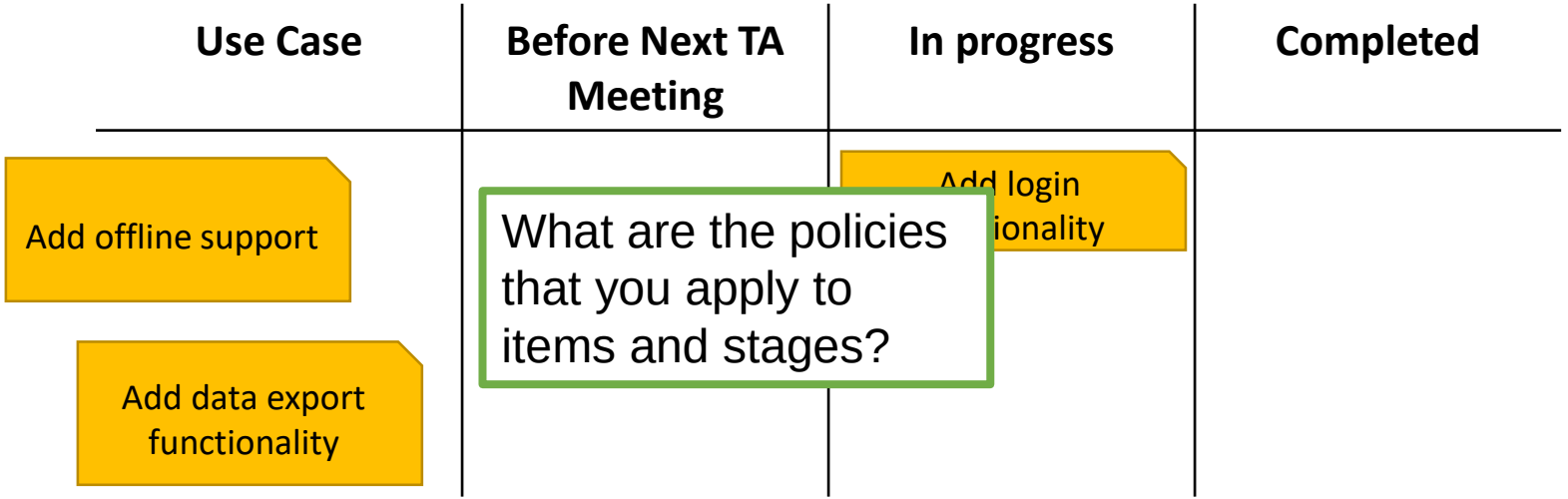
Definition of Workflow



Definition of Workflow



Definition of Workflow



Definition of Workflow