Agile Overview

Balachander Swaminathan (bala@thoughtworks.com)

Agenda for this session

- **The Story of Software Development**
- Lean Thinking
- Agile Values and Principles
- Agile Process
- Agile Practices
- Summary/Review
- Questions/Close

The Story of Software Development...

We started off with **Software Engineering...**

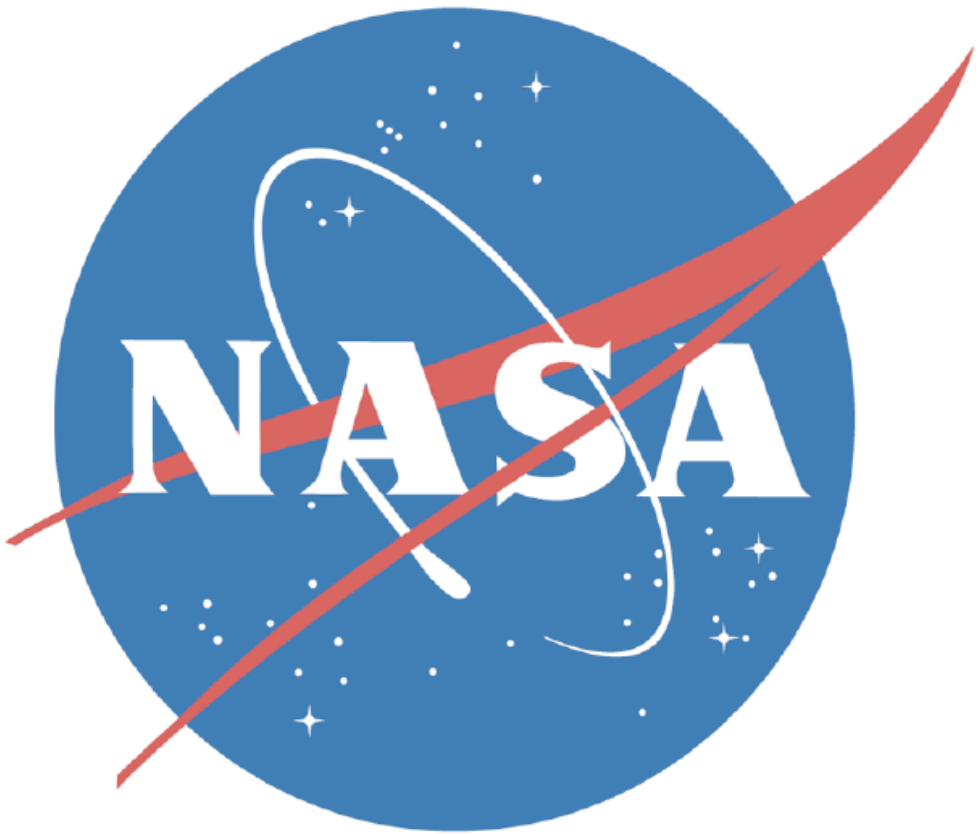
IEEE defines Software Engineering as....

"Software Engineering is the application of a systematic, disciplined, quantifiable approach to development, operation and maintenance of software: that is, the application of engineering to software."

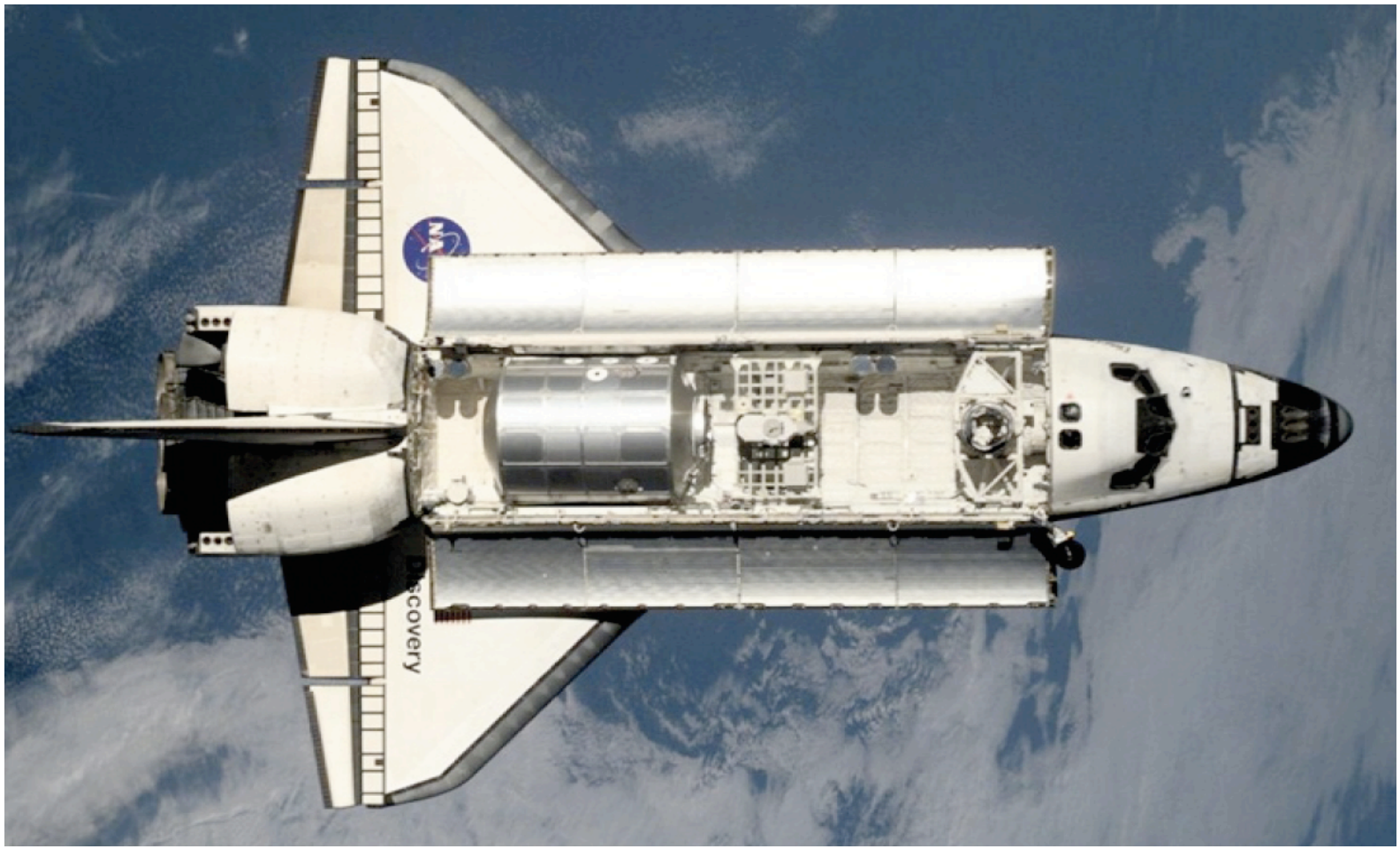
IEEE Standard Computer Dictionary,
ISBN 1-55937-079-3, 1990

Who does Software Engineering?

Who does Software Engineering?



For the space shuttle’s operating system



For the space shuttle's operating system

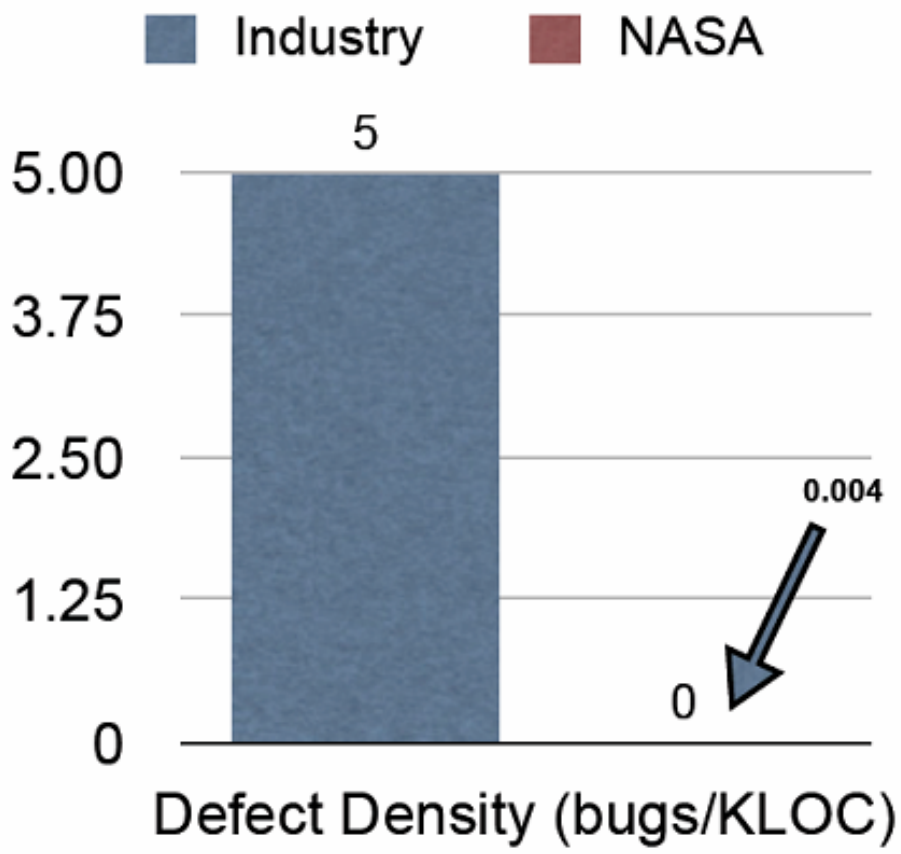
Some Statistics - NASA's Defect Density

Some Statistics - NASA's Defect Density

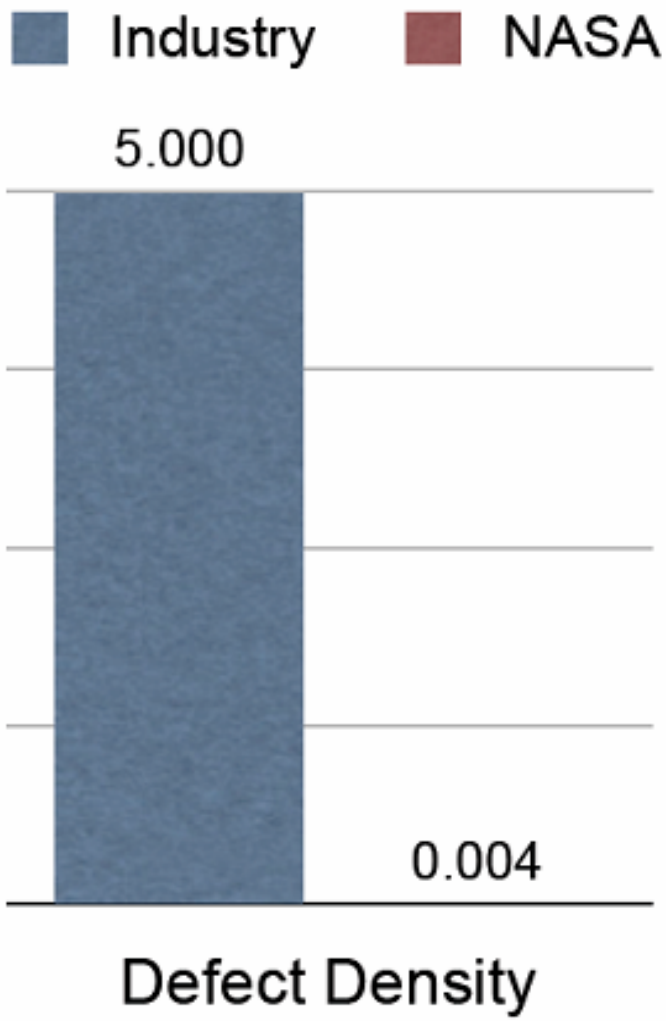
The last 11 versions of the space shuttle's 420,000 line systems had a total of 17 defects.

Some Statistics - NASA's Defect Density

The last 11 versions of the space shuttle's 420,000 line systems had a total of 17 defects.

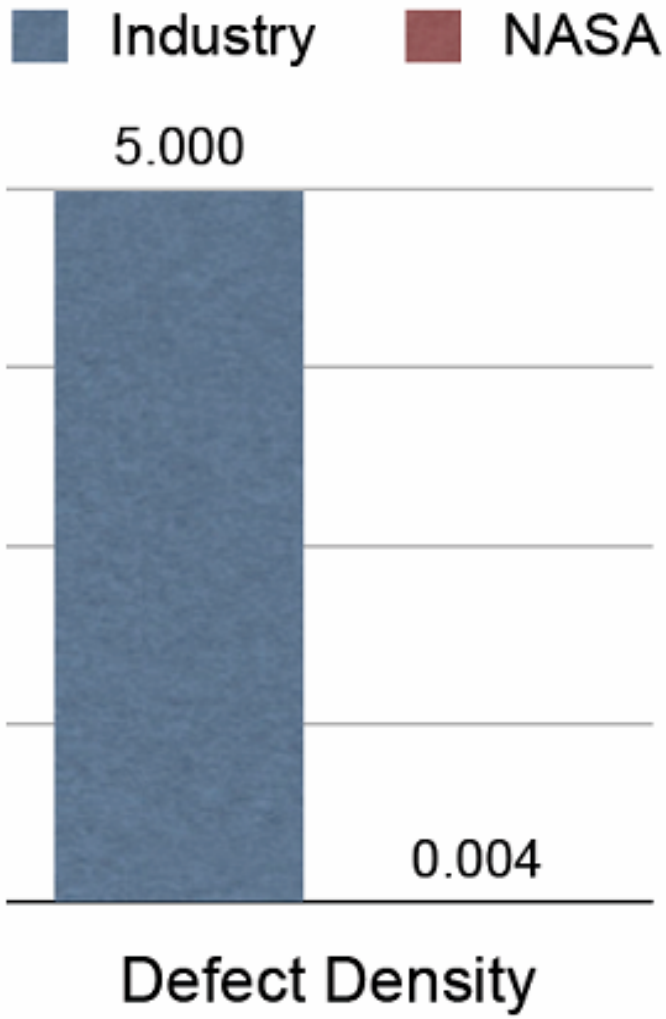


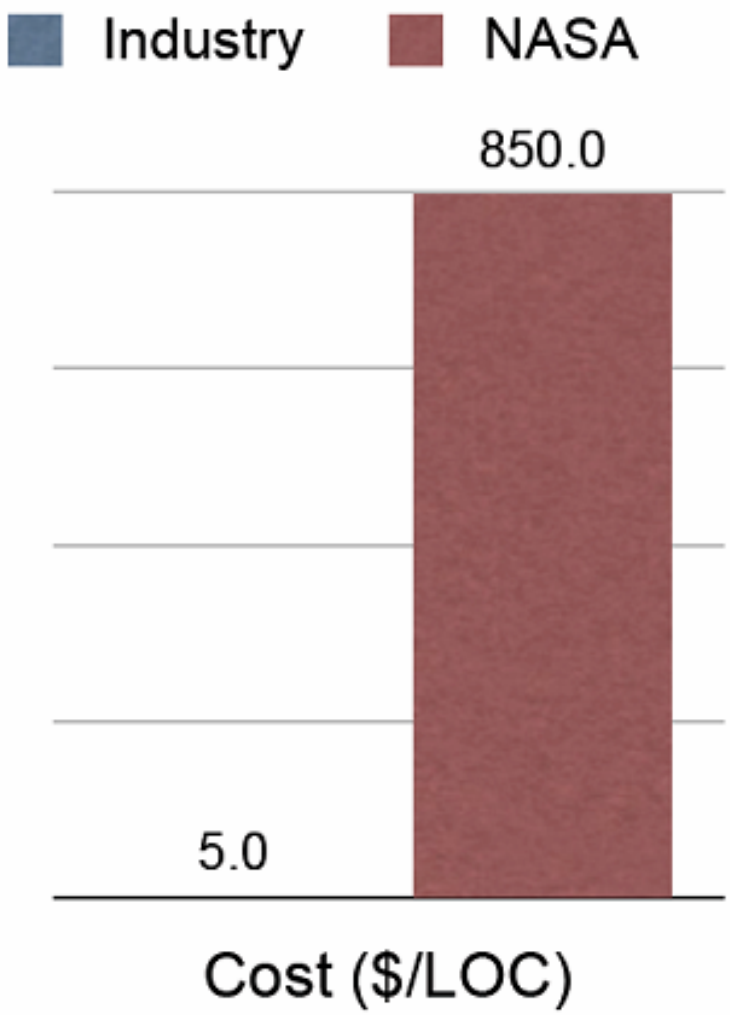
One More Data Point



8

One More Data Point





Another real software engineering project

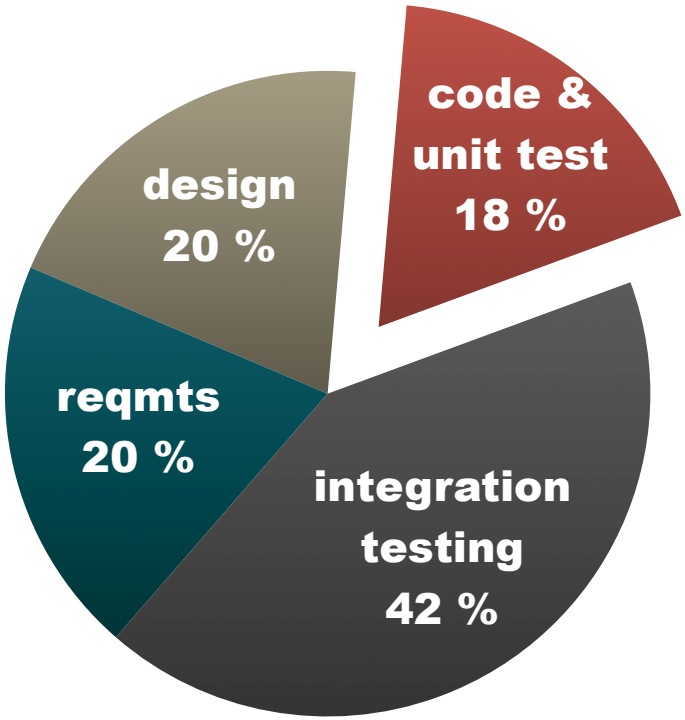
Another real software engineering project

Safeguard - Ballistic Missile Defense System

Another real software engineering project

Safeguard - Ballistic Missile Defense System

- 1969-1975, 5407 person years
- Hardware designed at the same time as software specs being written
- Late changes in requirements not an option

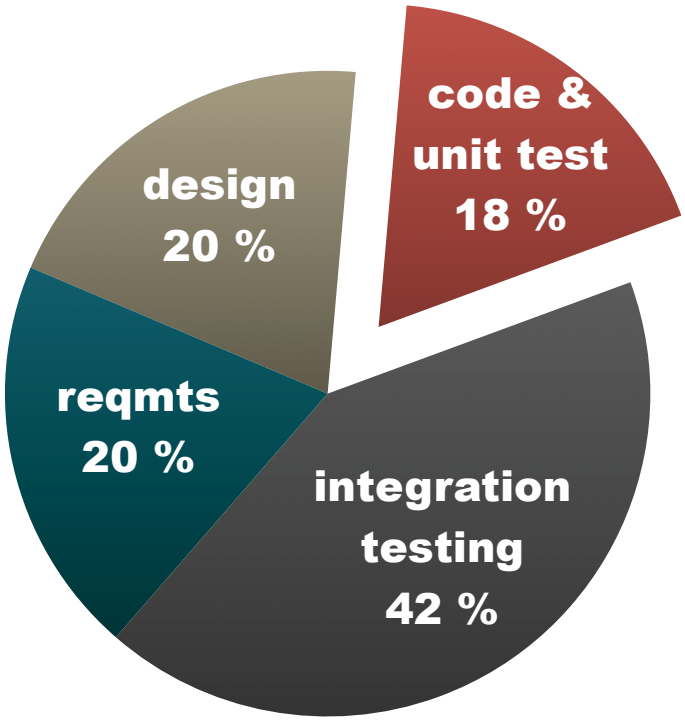


9

Another real software engineering project

Safeguard - Ballistic Missile Defense System

- 1969-1975, 5407 person years
- Hardware designed at the same time as software specs being written
- Late changes in requirements not an option



Did it Succeed?

9

Safeguard - Ballistic Missile Defense System...cont.

10

Safeguard - Ballistic Missile Defense System...cont.

Revised Project Statistics

10

Safeguard - Ballistic Missile Defense System...cont.

Revised Project Statistics

- The project was delivered according to specifications

10

Safeguard - Ballistic Missile Defense System...cont.

Revised Project Statistics

- The project was delivered according to specifications
- Cost: $25 Billion (not adjusted)

10

Safeguard - Ballistic Missile Defense System...cont.

Revised Project Statistics

- The project was delivered according to specifications
- Cost: $25 Billion (not adjusted)
- 1969-1975, 5407 person years

Safeguard - Ballistic Missile Defense System...cont.

Revised Project Statistics

- The project was delivered according to specifications
- Cost: $25 Billion (not adjusted)
- 1969-1975, 5407 person years

Operational for 133 days - Project terminated in 1978

10

Safeguard - Ballistic Missile Defense System...cont.

Revised Project Statistics

- The project was delivered according to specifications
- Cost: $25 Billion (not adjusted)
- 1969-1975, 5407 person years

Operational for 133 days - Project terminated in 1978

'By the time the 6-year anti-missile system project was completed, the new missiles were faster than the anti-missile missiles'

Where did things go wrong?

11

Where did things go wrong?

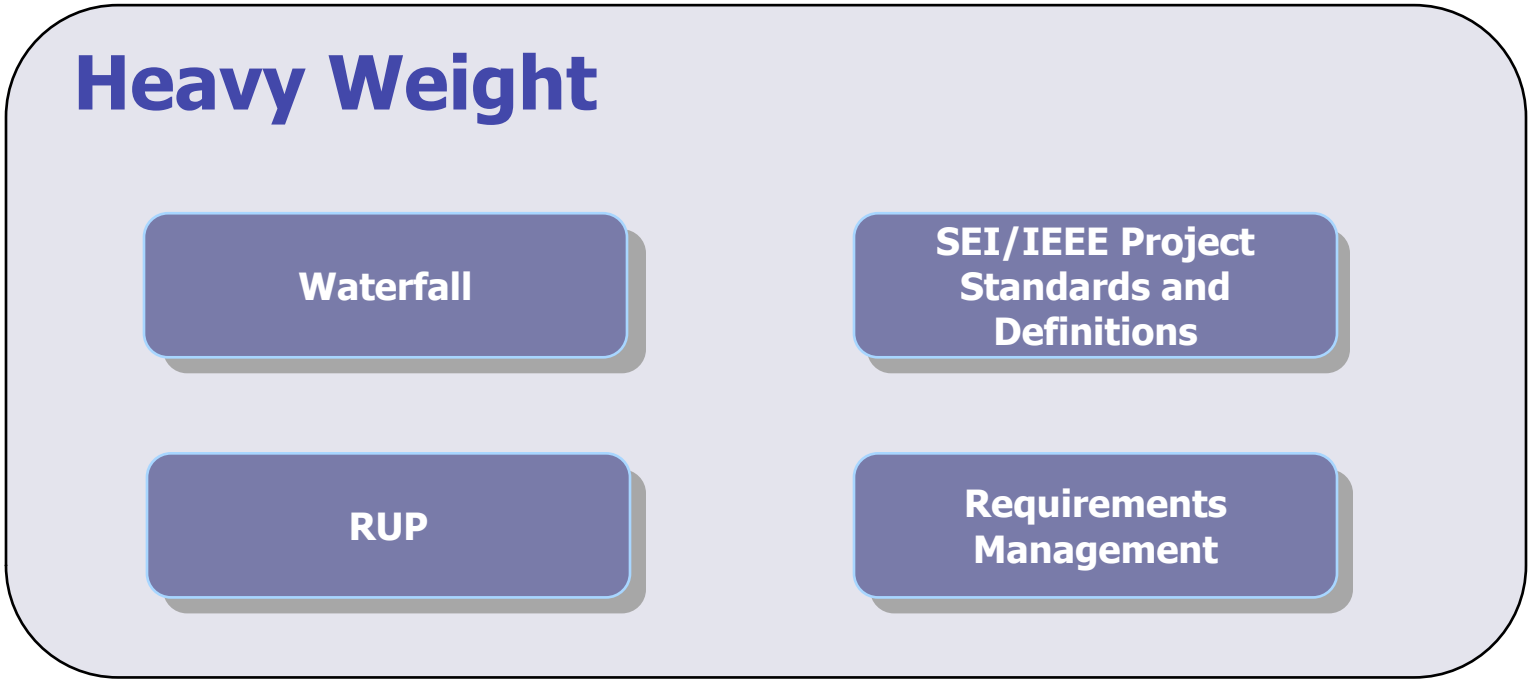
- Software Engineering is a heavy weight methodology and such heavy weight methodologies characteristically are most successful when:
  - Requirements are **stable**
  - **Technology** is well known and mature
  - Everything happens **as one would expect**
  - We are **not taking** on anything **new** or **unknown**
  - We have done this **many times before**

Where did things go wrong?

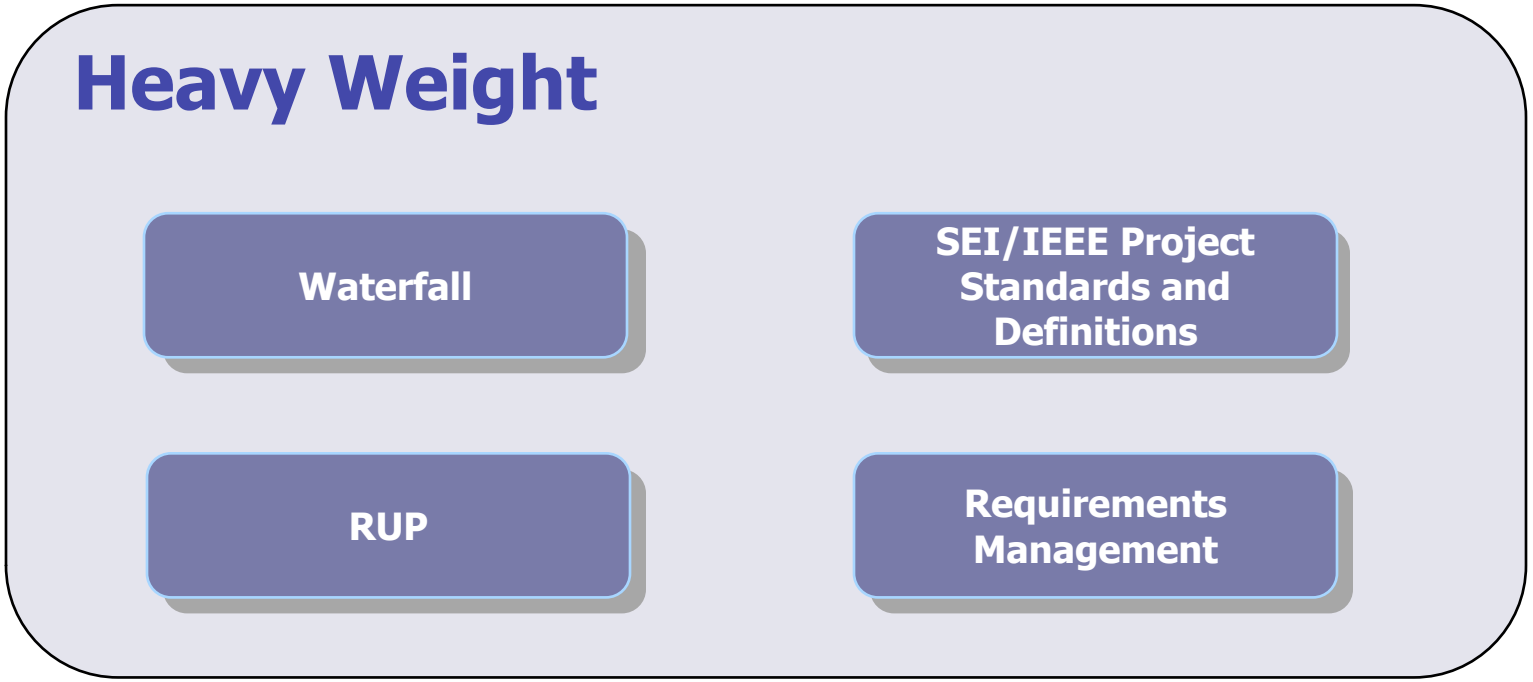
- Software Engineering is a heavy weight methodology and such heavy weight methodologies characteristically are most successful when:
  - Requirements are **stable**
  - **Technology** is well known and mature
  - Everything happens **as one would expect**
  - We are **not taking** on anything **new** or **unknown**
  - We have done this **many times before**

Projects with these characteristics are few and far between.

Other Heavy Weight Methodologies



Other Heavy Weight Methodologies



Heavy weight methodologies work in some instances, but there are **high costs,** and the **risk** in using them in **dynamic environments** is **high**.

12

So, heavy weight methodologies don't seem to meet our needs

Is there an alternative?

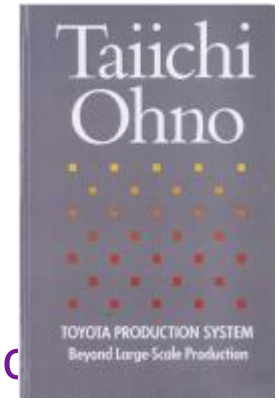
Agenda for this session

- The Story of Software Development
- **Lean Thinking**
- Agile Values and Principles
- Agile Process
- Agile Practices
- Summary/Review
- Questions/Close

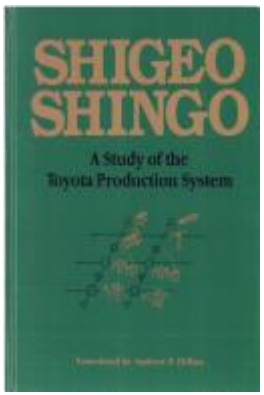
14

Lean Thinking – Eliminate Waste

- **The Toyota Production System,** 1988 (1978), Taichii Ohno
  - Pull Scheduling - Just-in-Time Flow
  - Expose Problems - Stop-the-Line Culture

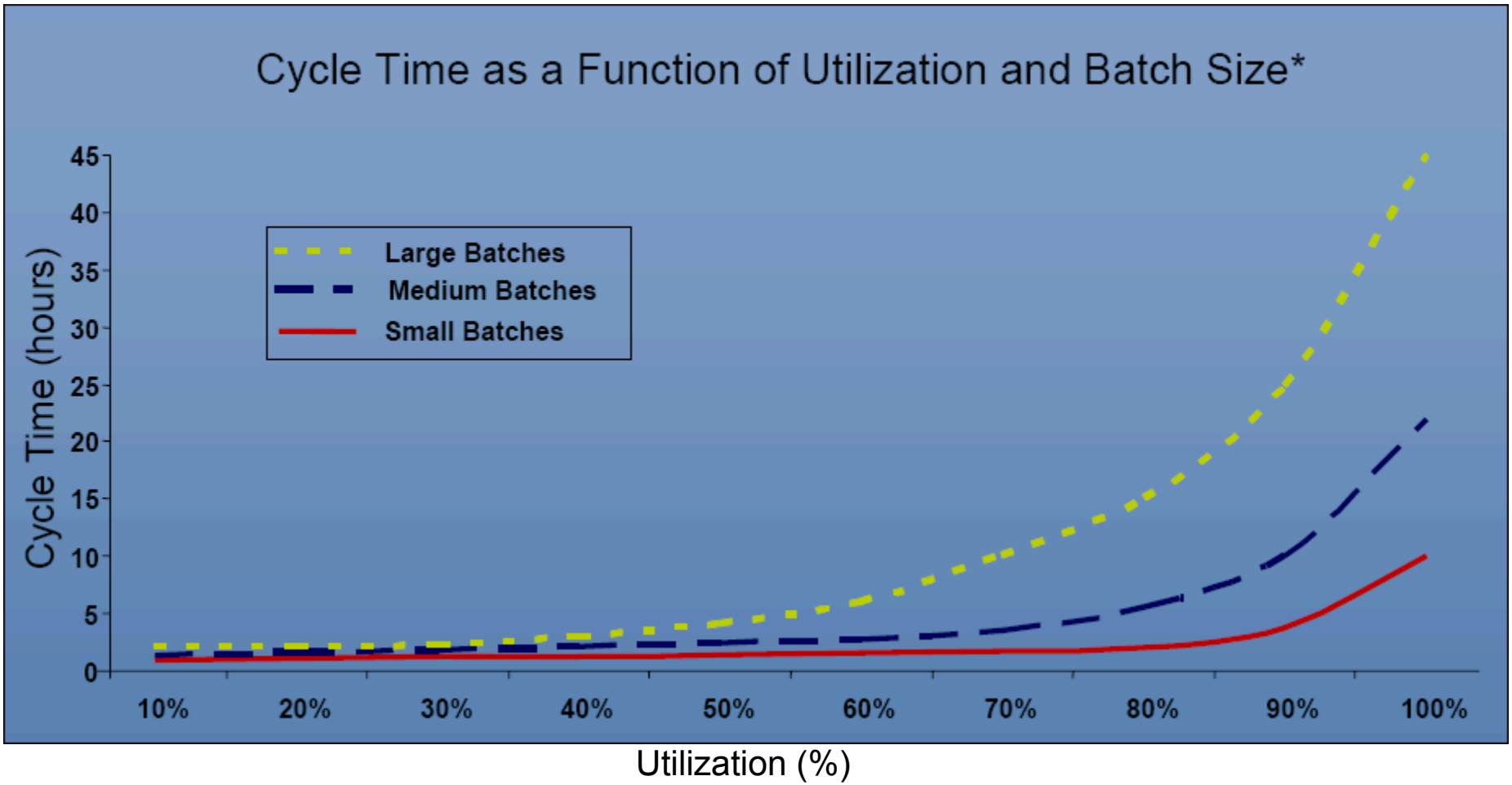


- **Study Of 'Toyota' Production System,** 1981, Shigeo Shingo
  - Non-Stock Production - Single Minute Setup
  - Zero Inspection – Automatic Error Detection at Every Step



15

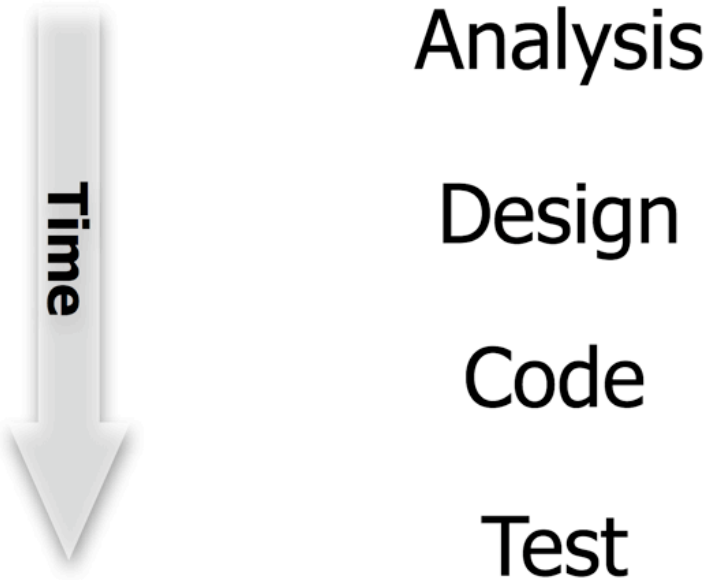
Lessons from Queuing theory



Source: Beyond Agile Software Development Becoming Lean, Mary Poppendieck, Poppendieck.llc

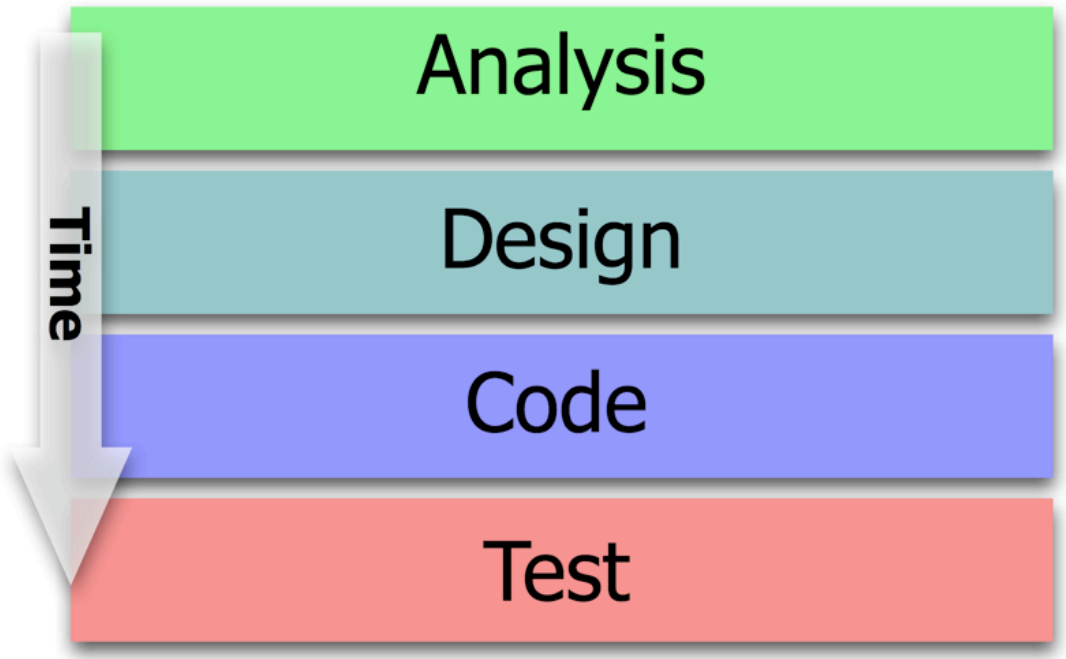
Applying Lean Principles to Software Development

Traditional Process



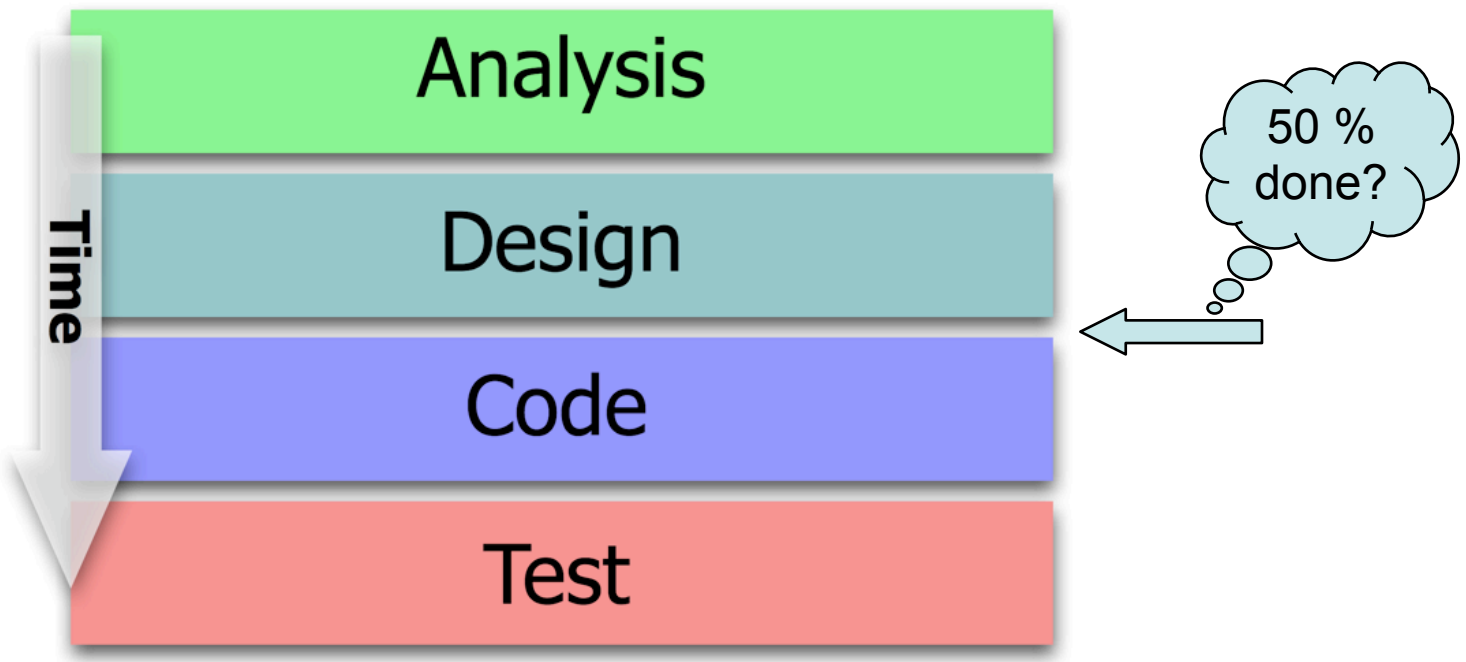
Applying Lean Principles to Software Development

Traditional Process



Applying Lean Principles to Software Development

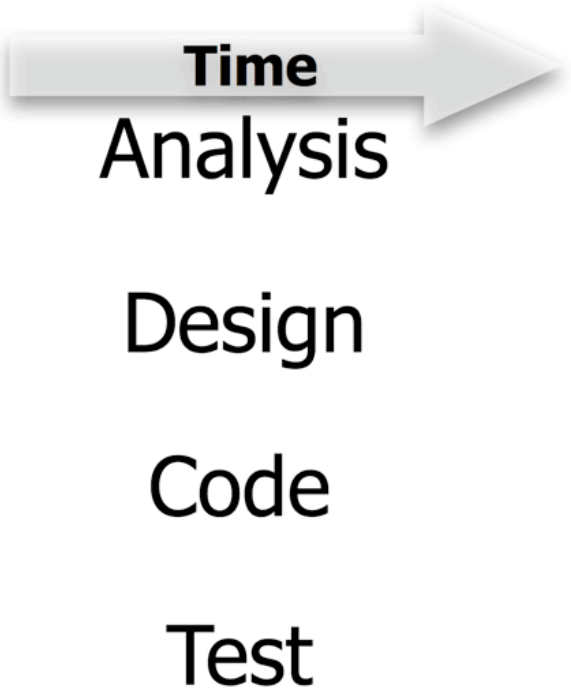
Traditional Process



17

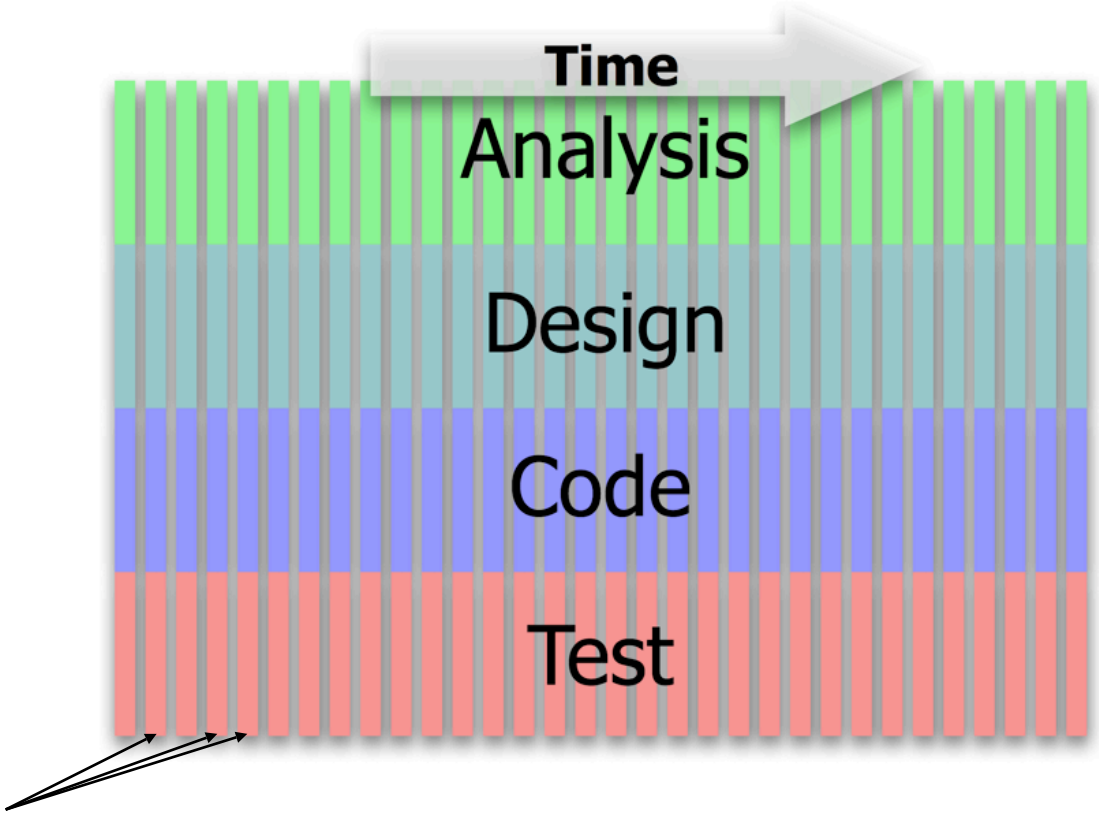
Applying Lean Principles to Software Development…cont.

A better way of doing the same



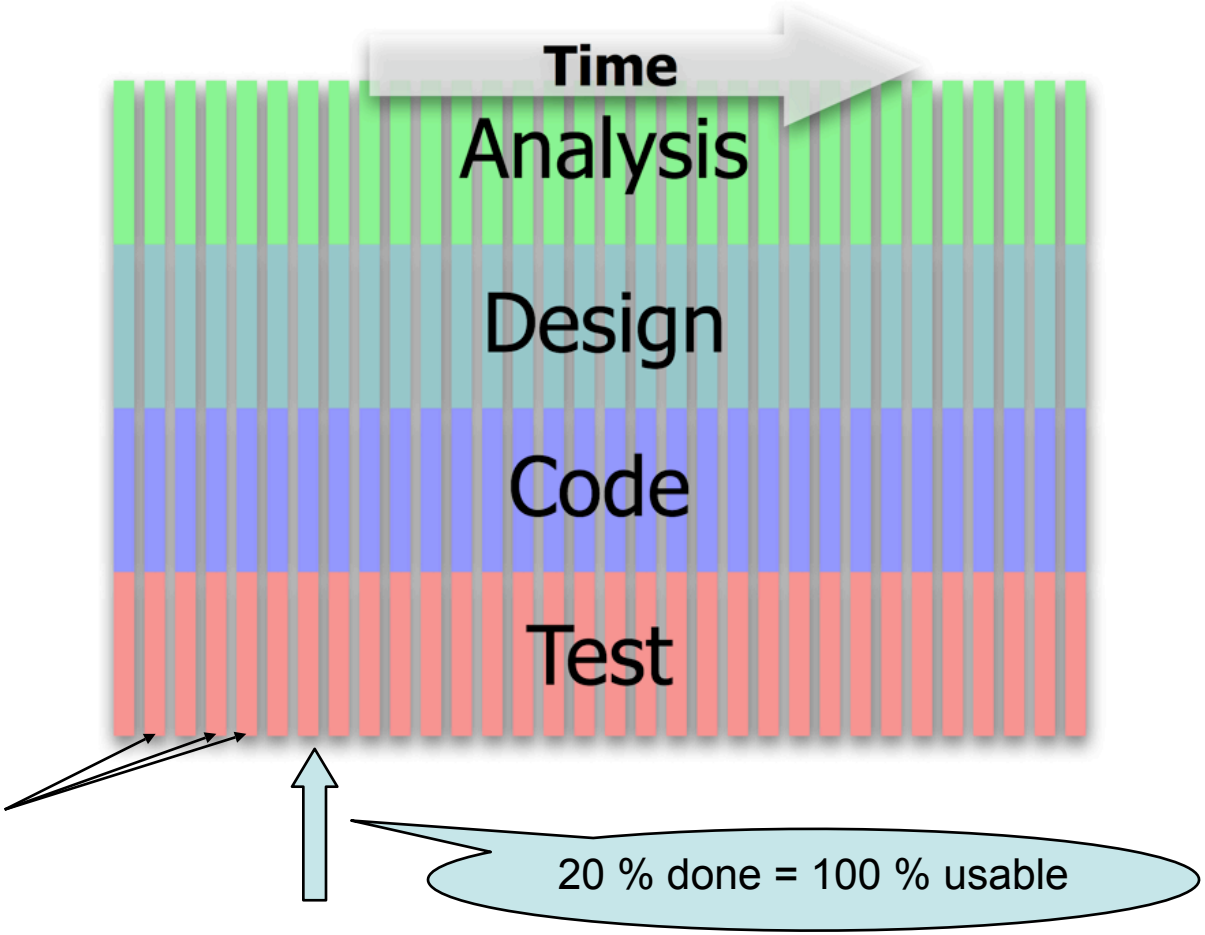
Applying Lean Principles to Software Development…cont.

A better way of doing the same



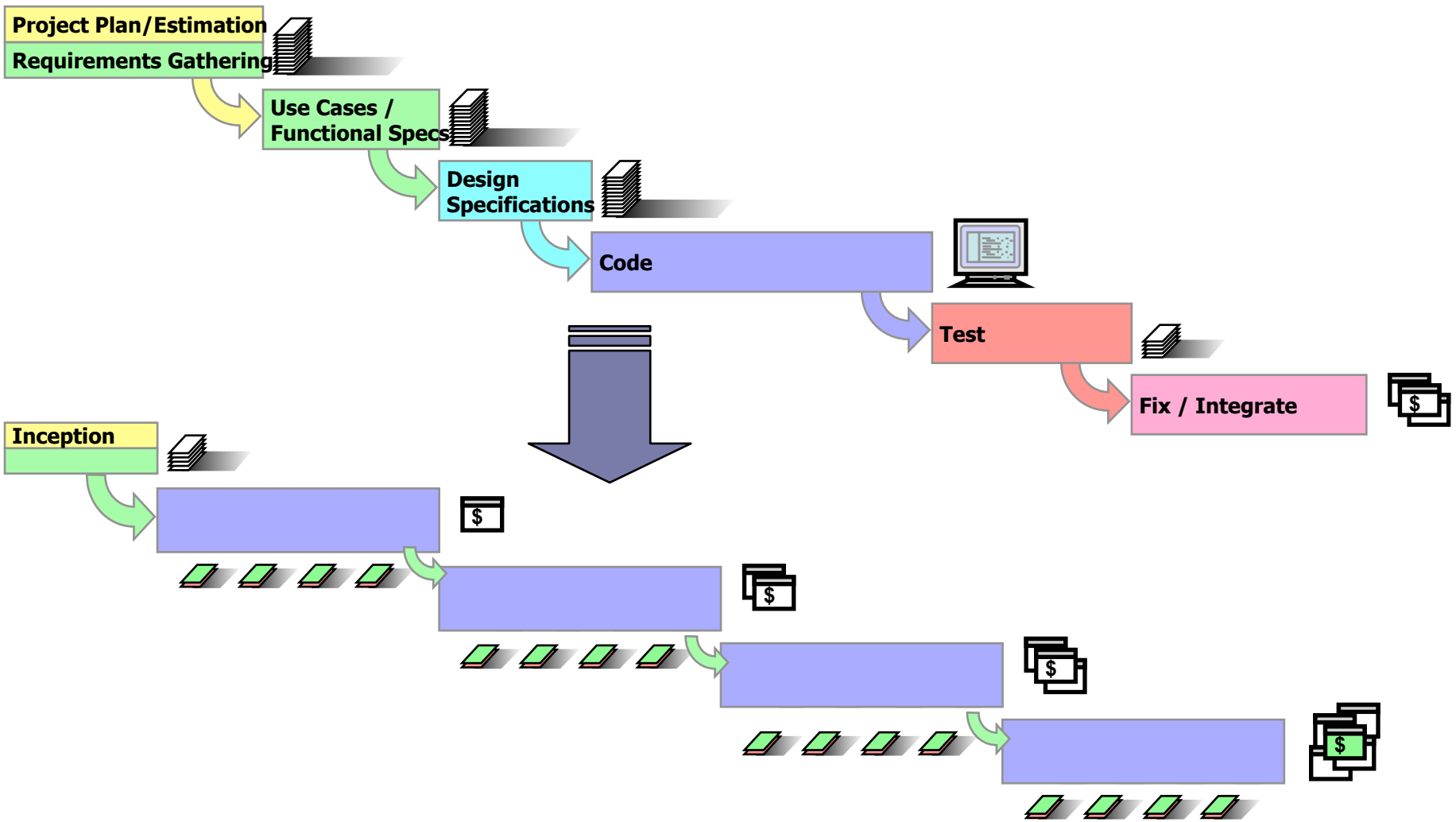
Applying Lean Principles to Software Development…cont.

A better way of doing the same



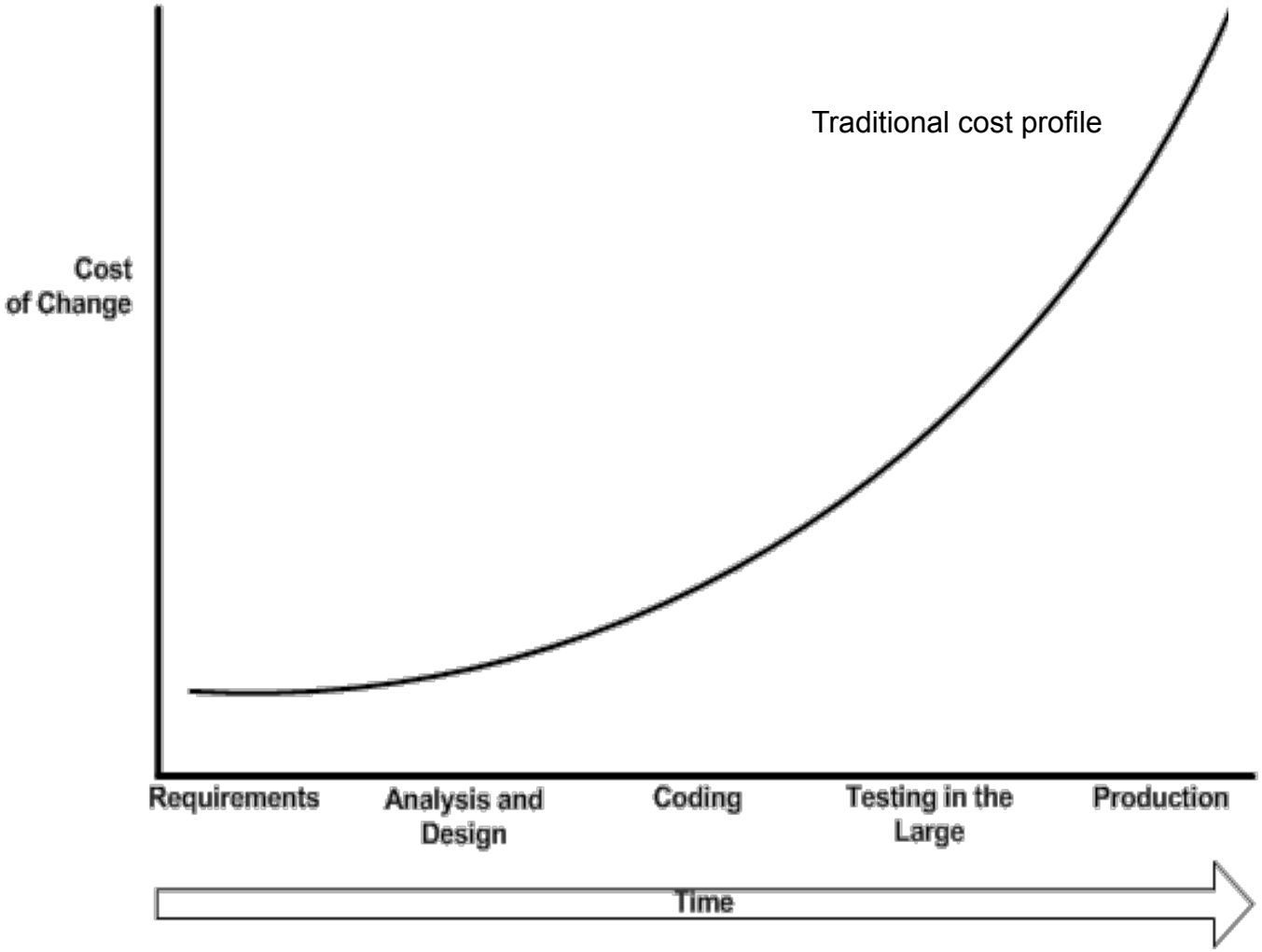
18

Lean Principles applied to Software Development



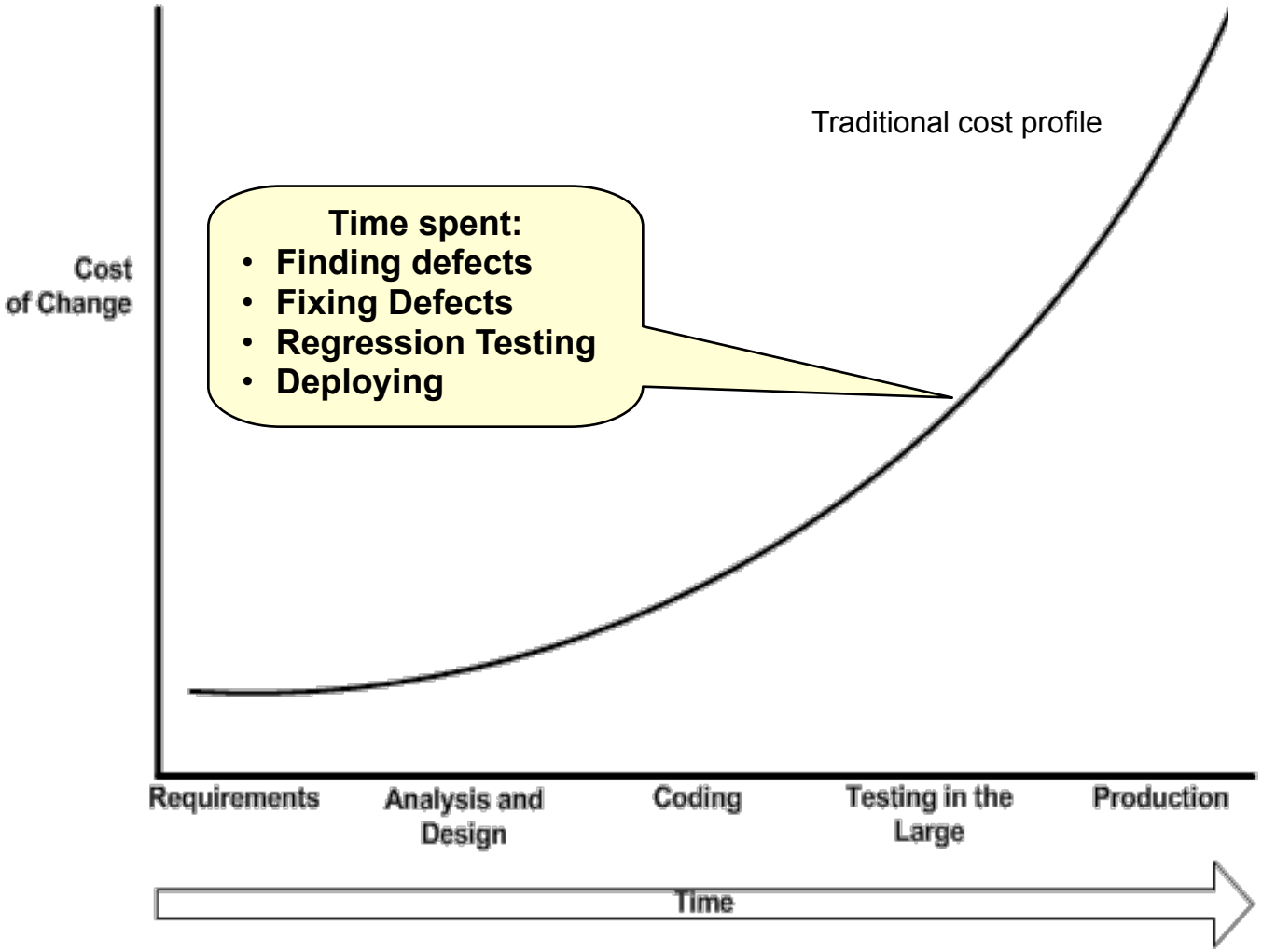
19

Lower cost of change through higher quality software



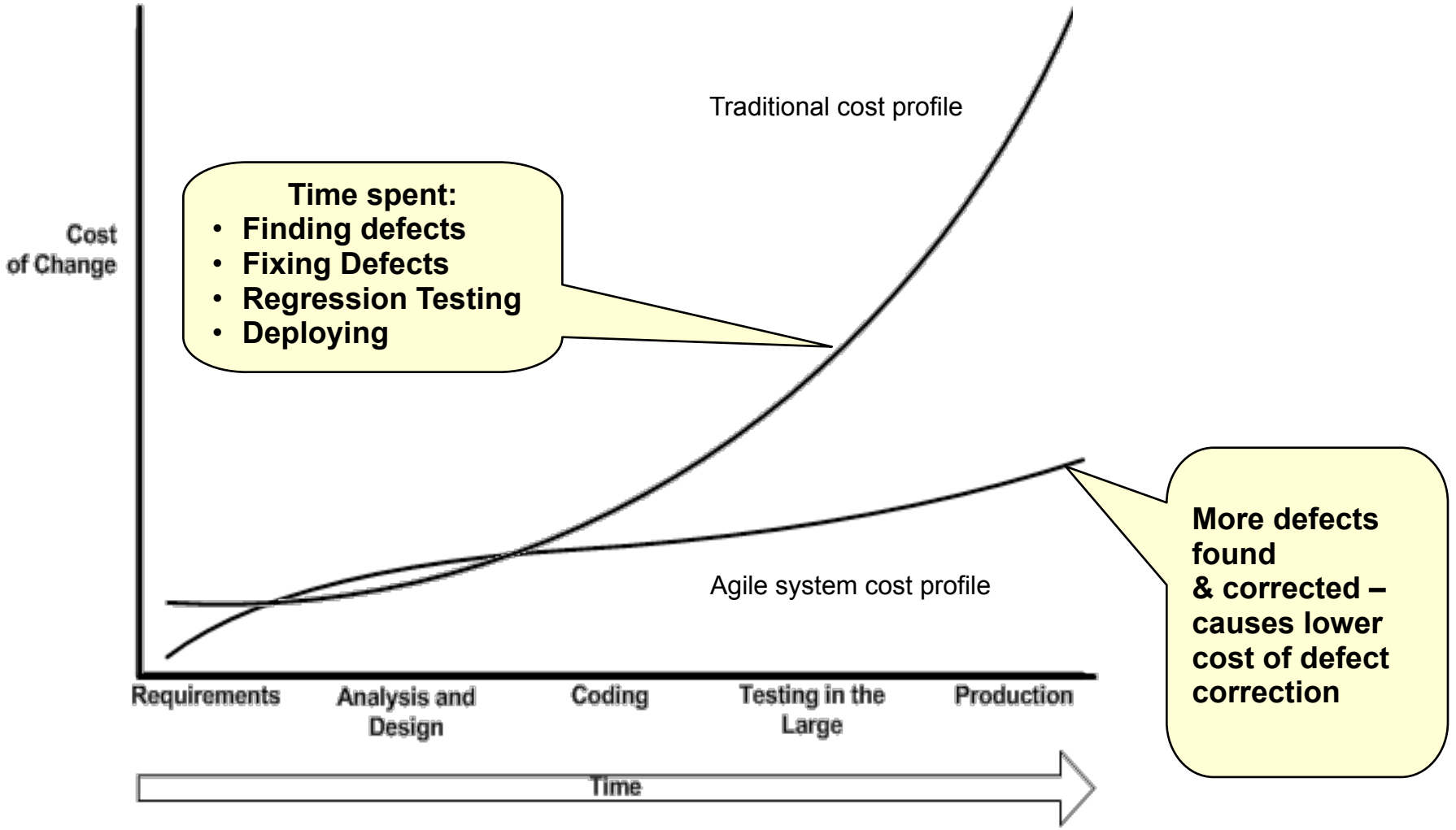
20

Lower cost of change through higher quality software



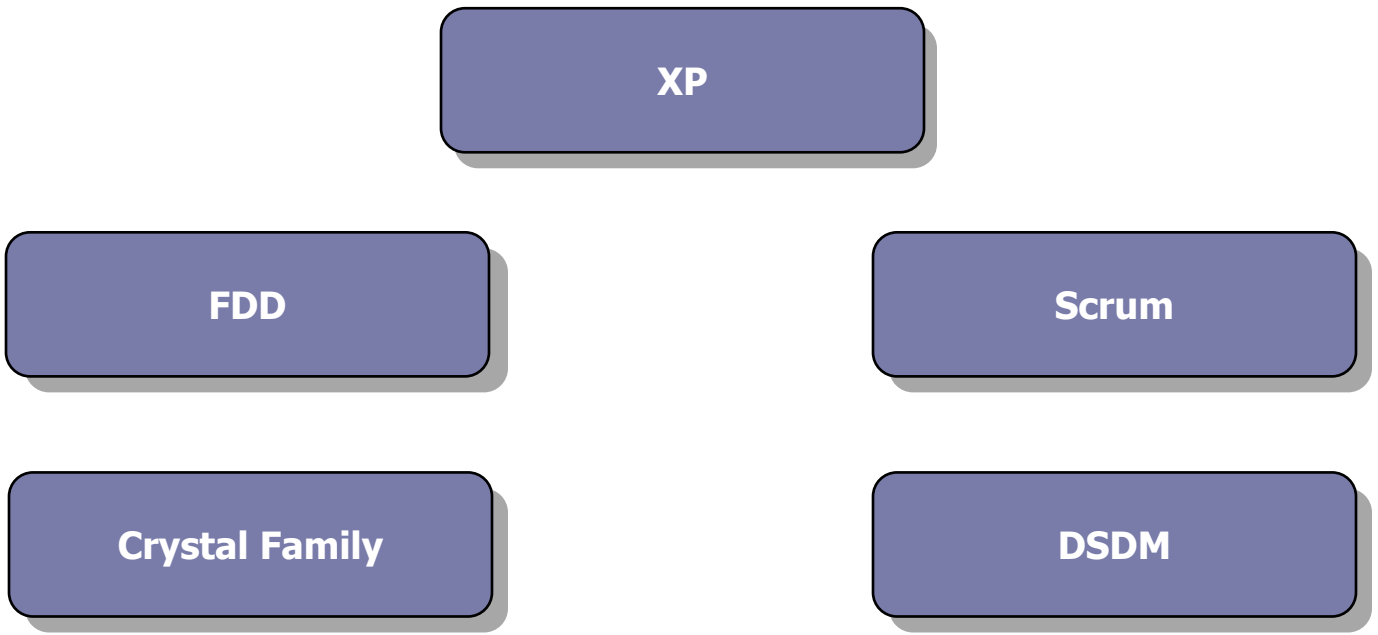
20

Lower cost of change through higher quality software



20

New Methodologies Emerged



Agenda for this session

- The Story of Software Development
- Lean Thinking
- **Agile Values and Principles**
- Agile Process
- Agile Practices
- Summary/Review
- Questions/Close

22

23

2000

23

2000

XP | Extreme Programming (Kent Beck)

DSDM | Dynamic System Development Method (Dane Faulkner)

FDD | Feature Driven Development (Jeff DeLuca)

SCRUM (Ken Schwaber)

Crystal (Alistair Cockburn)

Adaptive Software Development (Jim Highsmith)

Lean Software Development (Mary Poppendieck)

2001

23

Agile manifesto

23

Agile manifesto

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

23

Agile manifesto

"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.

23

Agile manifesto

"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.

23

Agile manifesto

"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.

23

Agile manifesto

"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.

23

Agile manifesto

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

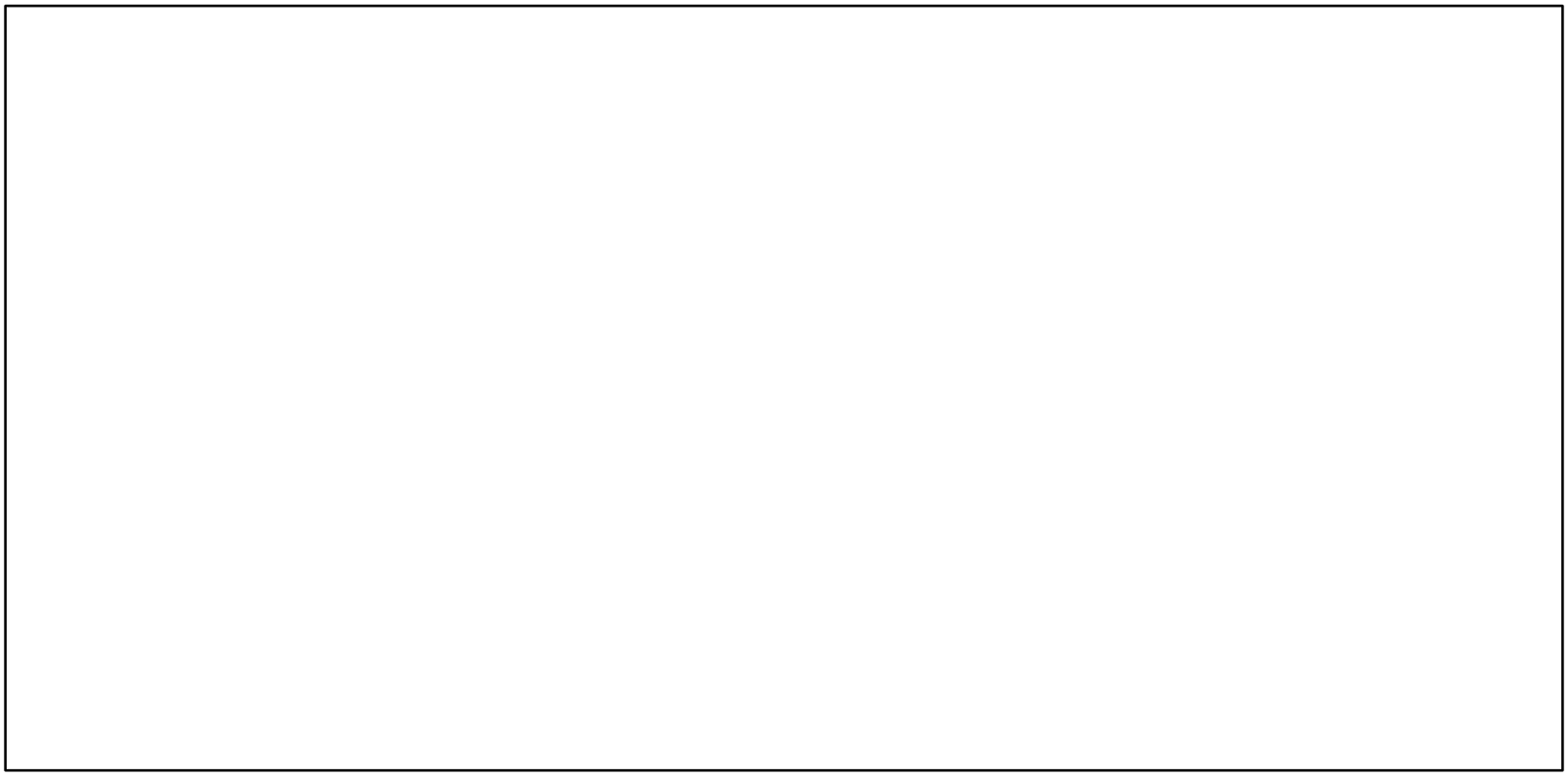
- **Individuals and interactions** over processes and tools.
- **Working software** over comprehensive documentation.
- **Customer collaboration** over contract negotiation.
- **Responding to change** over following a plan.

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

Agile (XP) Values

24

Agile (XP) Values



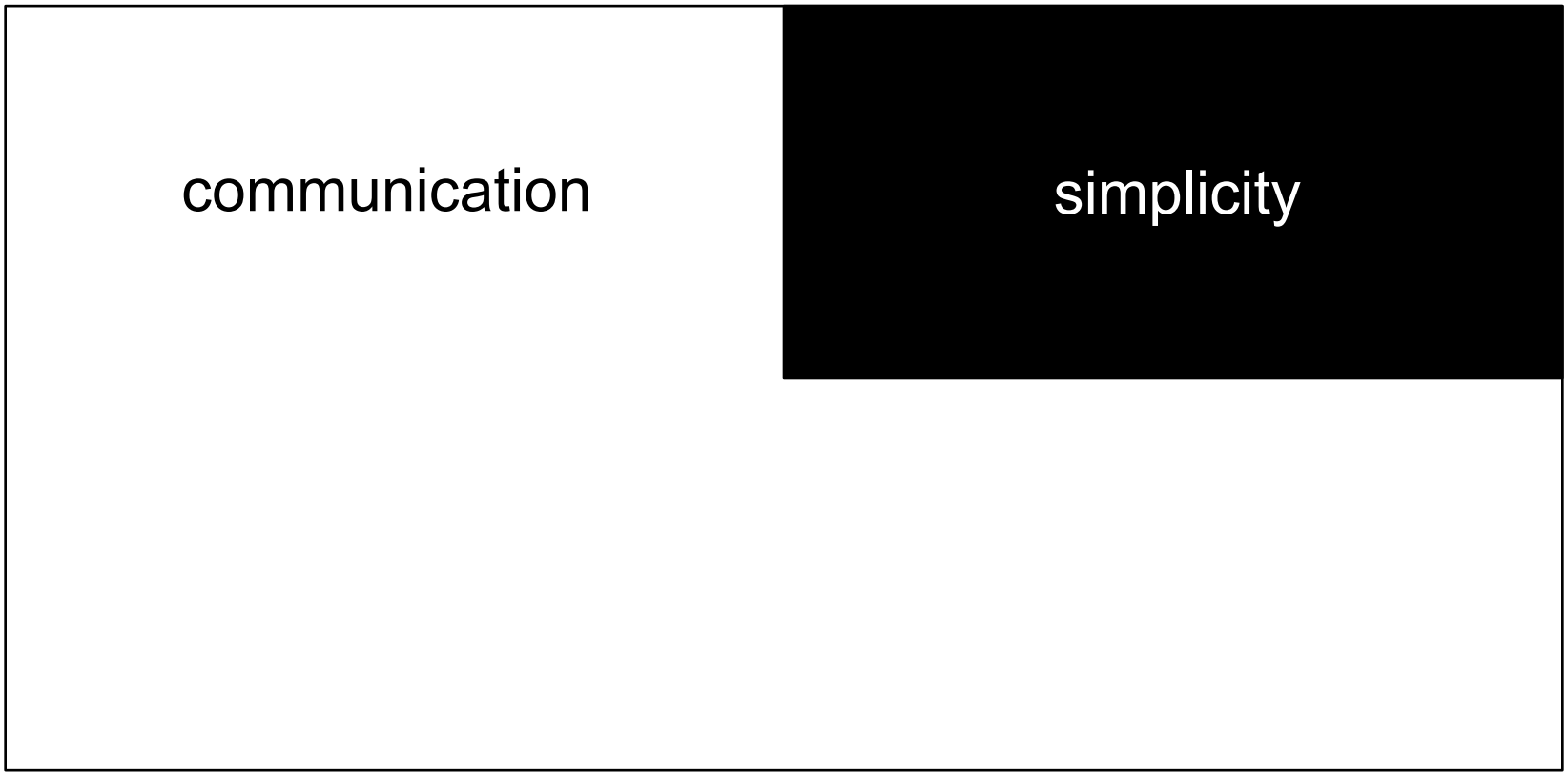
24

Agile (XP) Values

communication

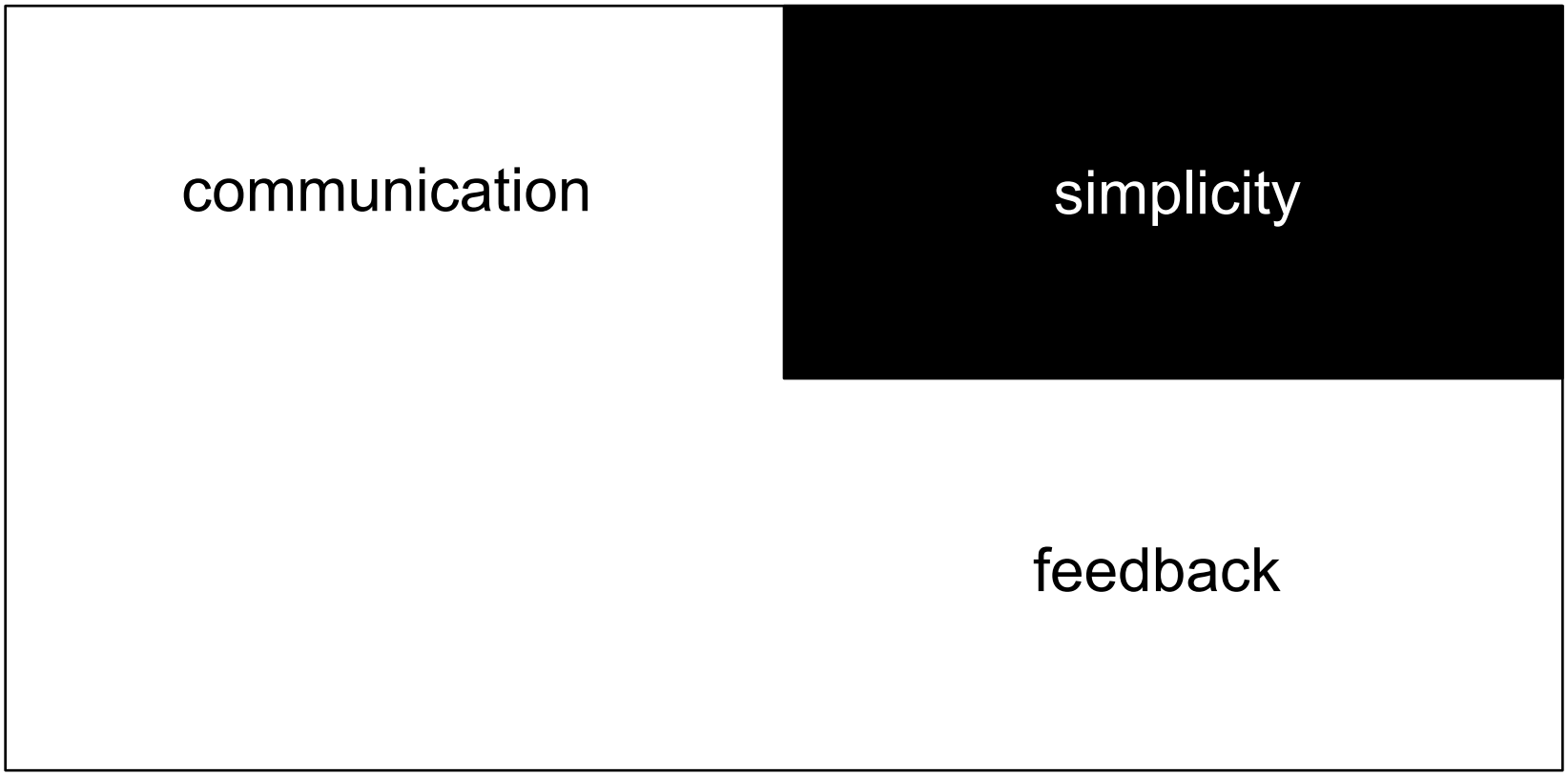
24

Agile (XP) Values



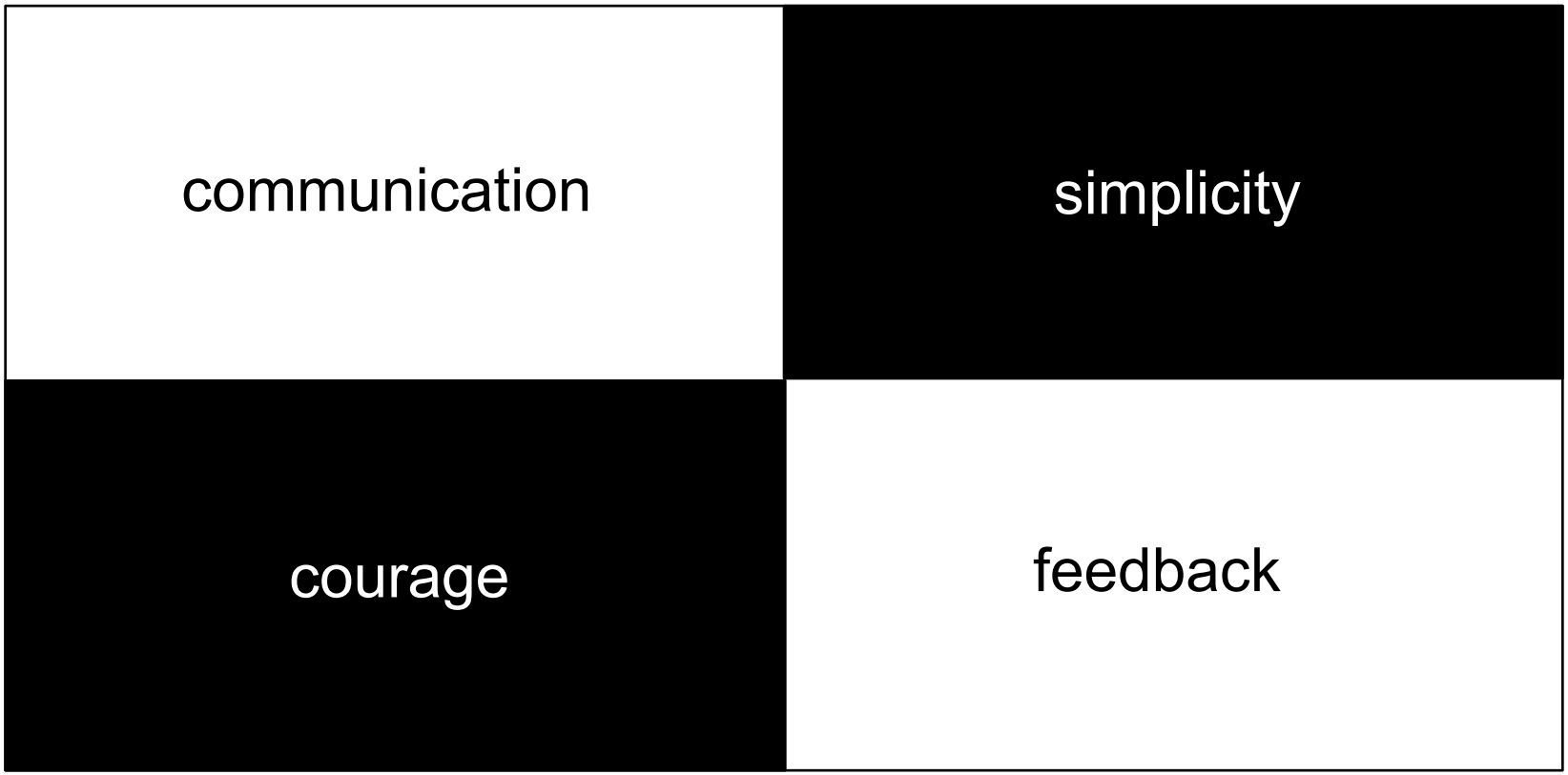
24

Agile (XP) Values



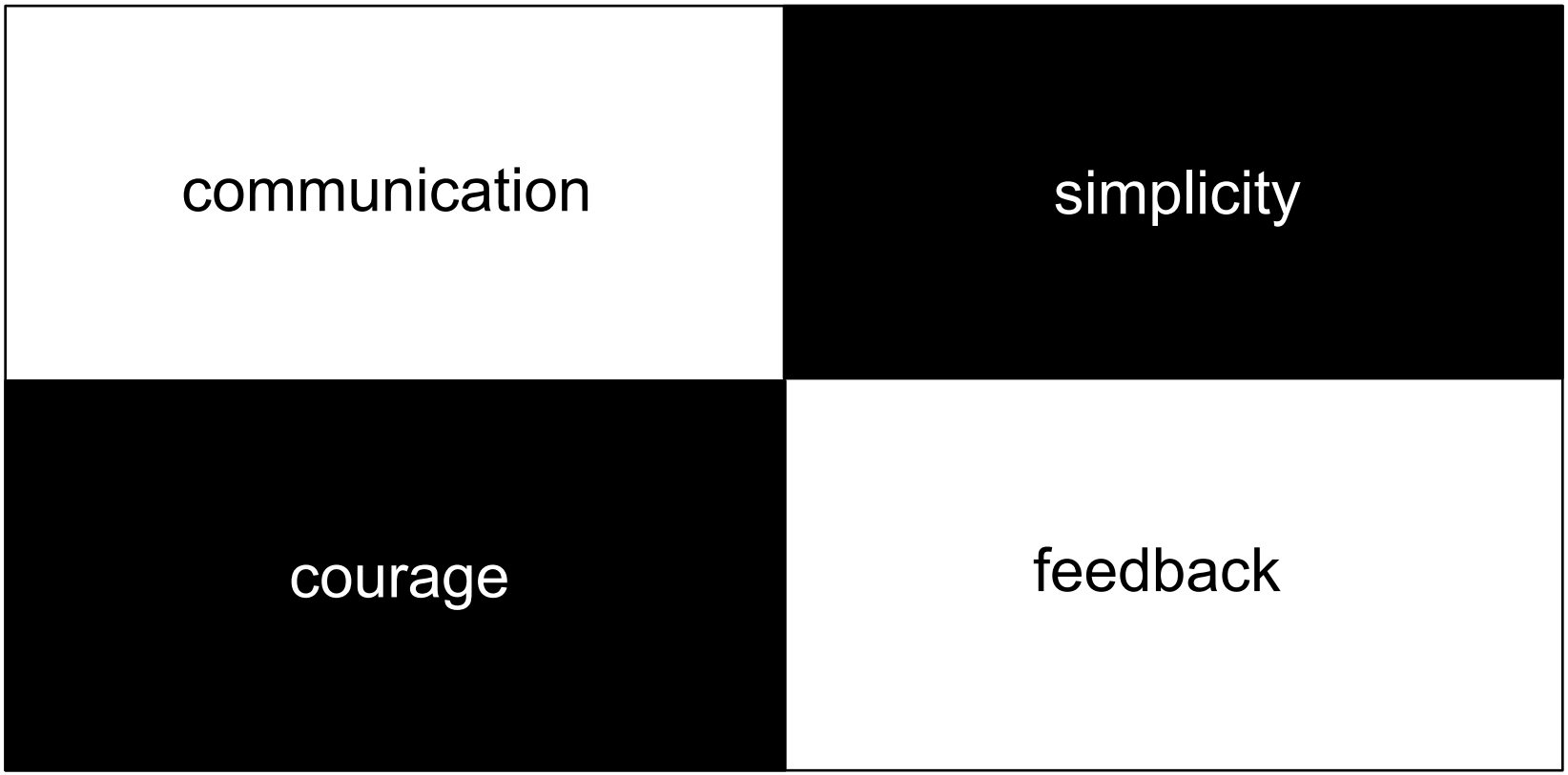
24

Agile (XP) Values



24

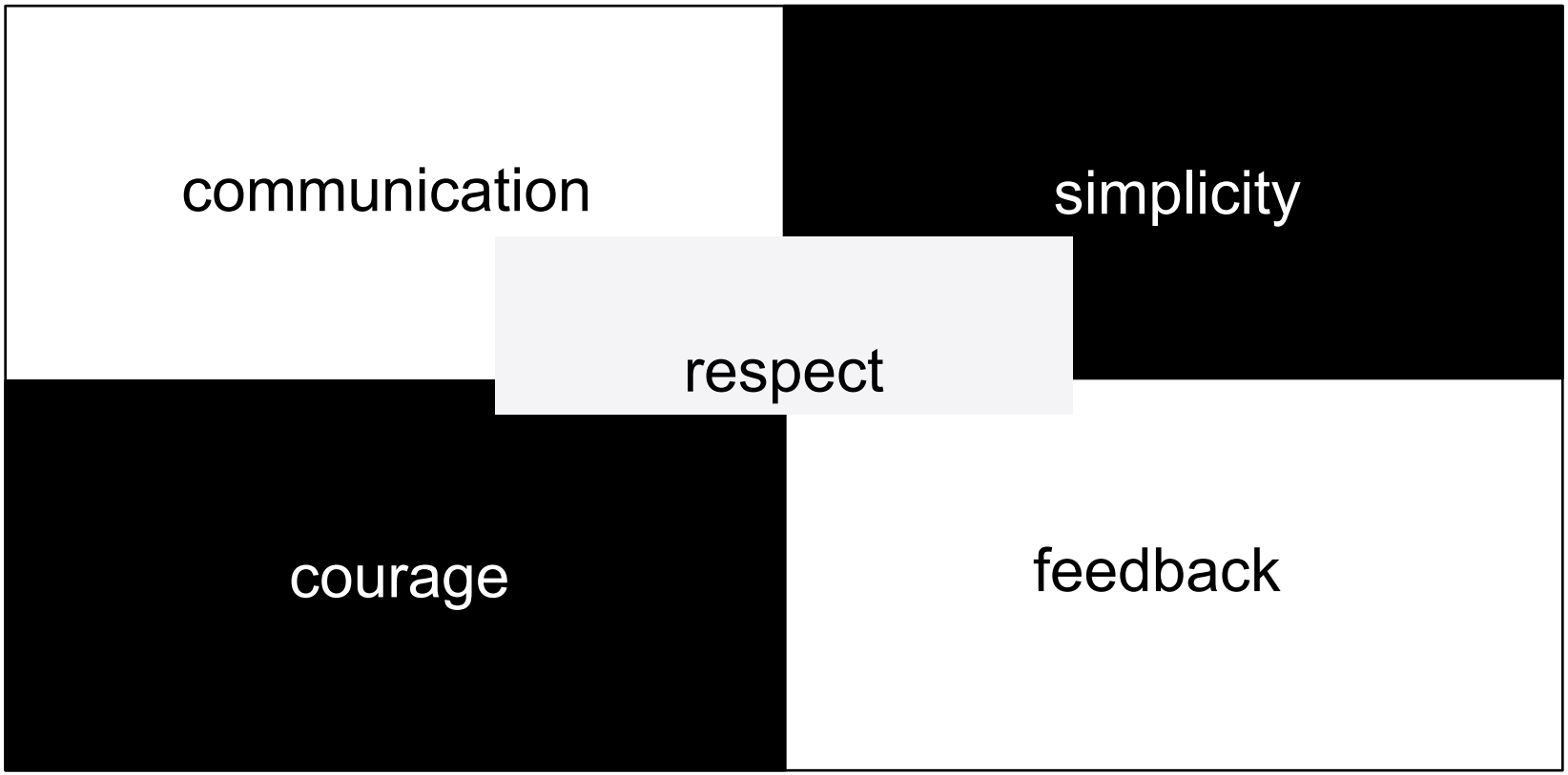
Agile (XP) Values



Communication leads to valuable feedback which encourages simplicity which allows for courage to change

24

Agile (XP) Values



Communication leads to valuable feedback which encourages simplicity which allows for courage to change

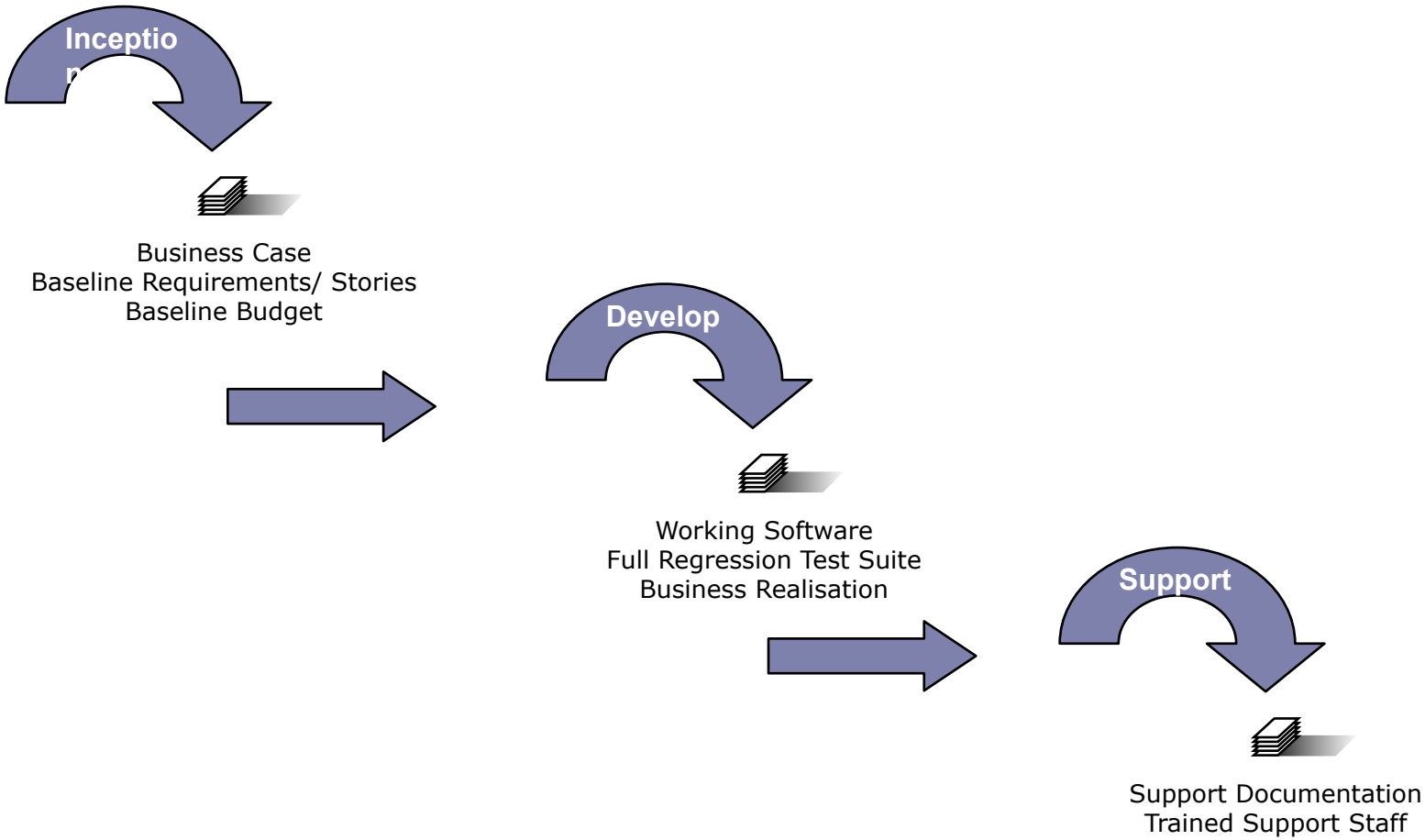
24

Agenda for this session

- The Story of Software Development
- Lean Thinking
- Agile Values and Principles
- **Agile Process**
- Agile Practices
- Summary/Review
- Questions/Close

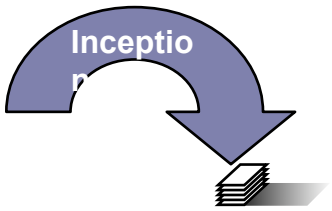
25

How Agile fits into software delivery



26

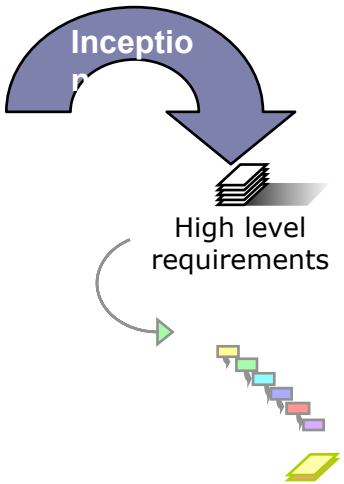
Agile development is an iterative and incremental process



High level requirements

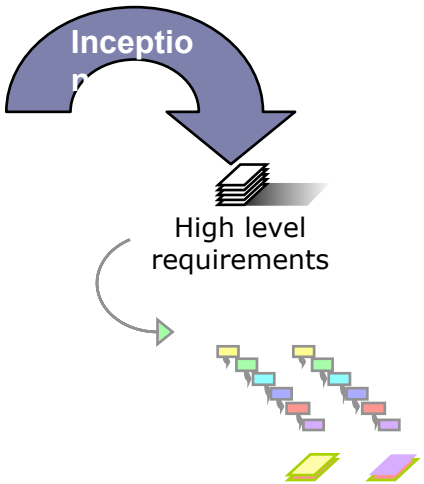
27

Agile development is an iterative and incremental process



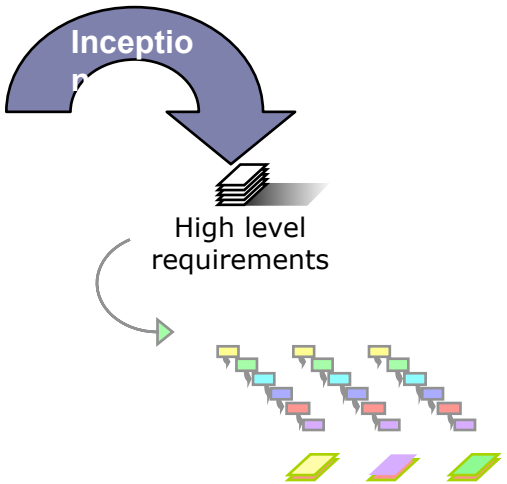
27

Agile development is an iterative and incremental process



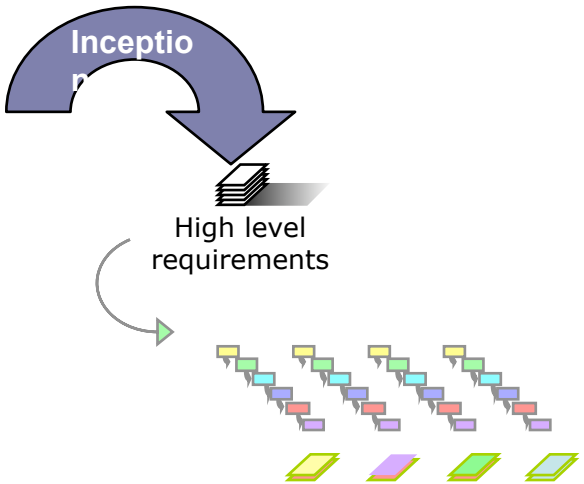
27

Agile development is an iterative and incremental process



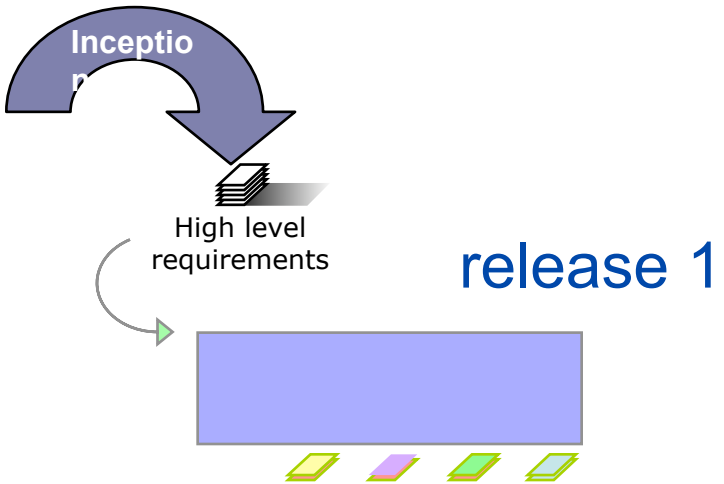
27

Agile development is an iterative and incremental process



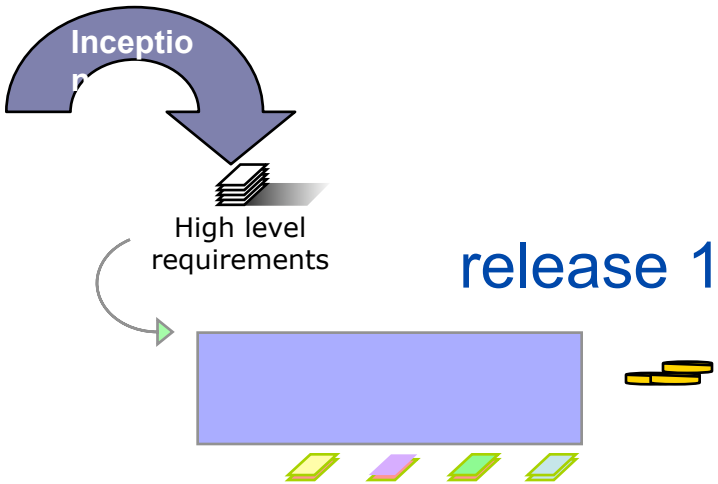
27

Agile development is an iterative and incremental process



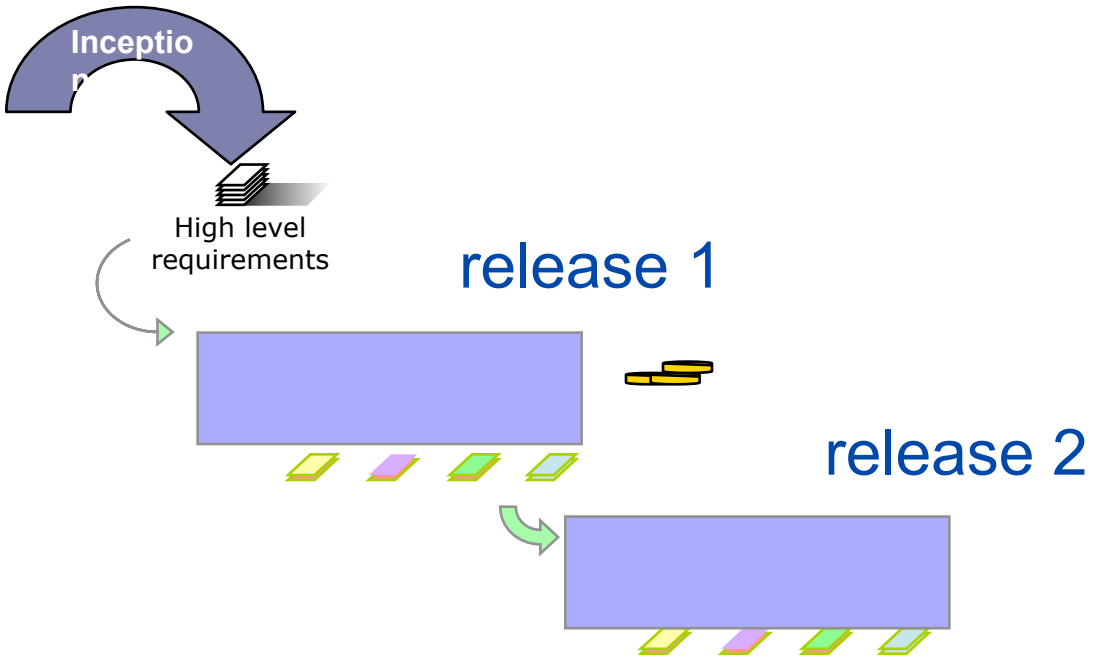
27

Agile development is an iterative and incremental process



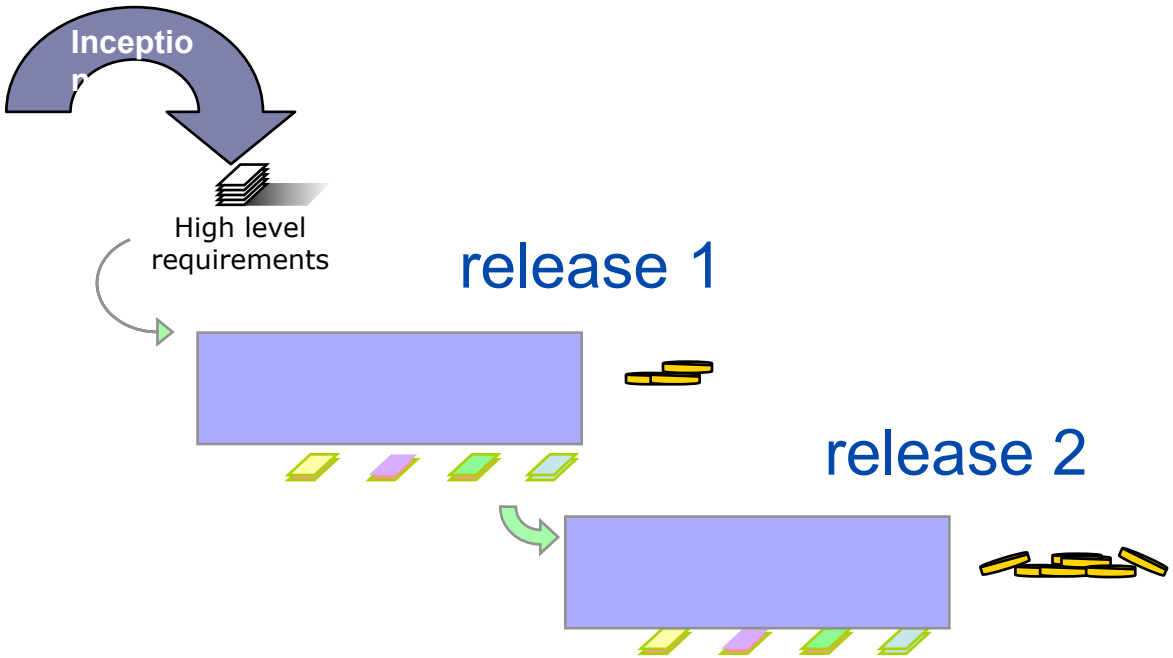
27

Agile development is an iterative and incremental process



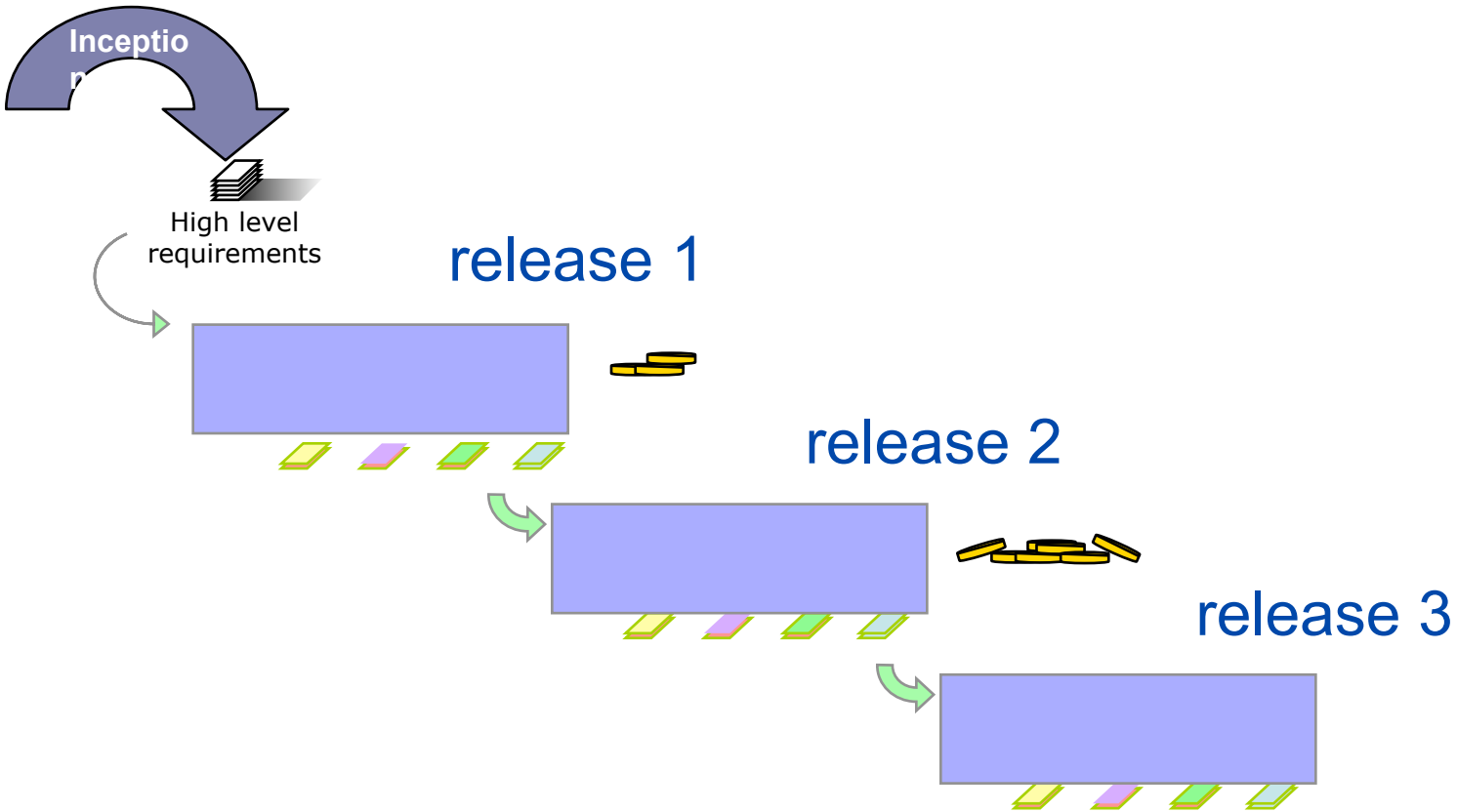
27

Agile development is an iterative and incremental process



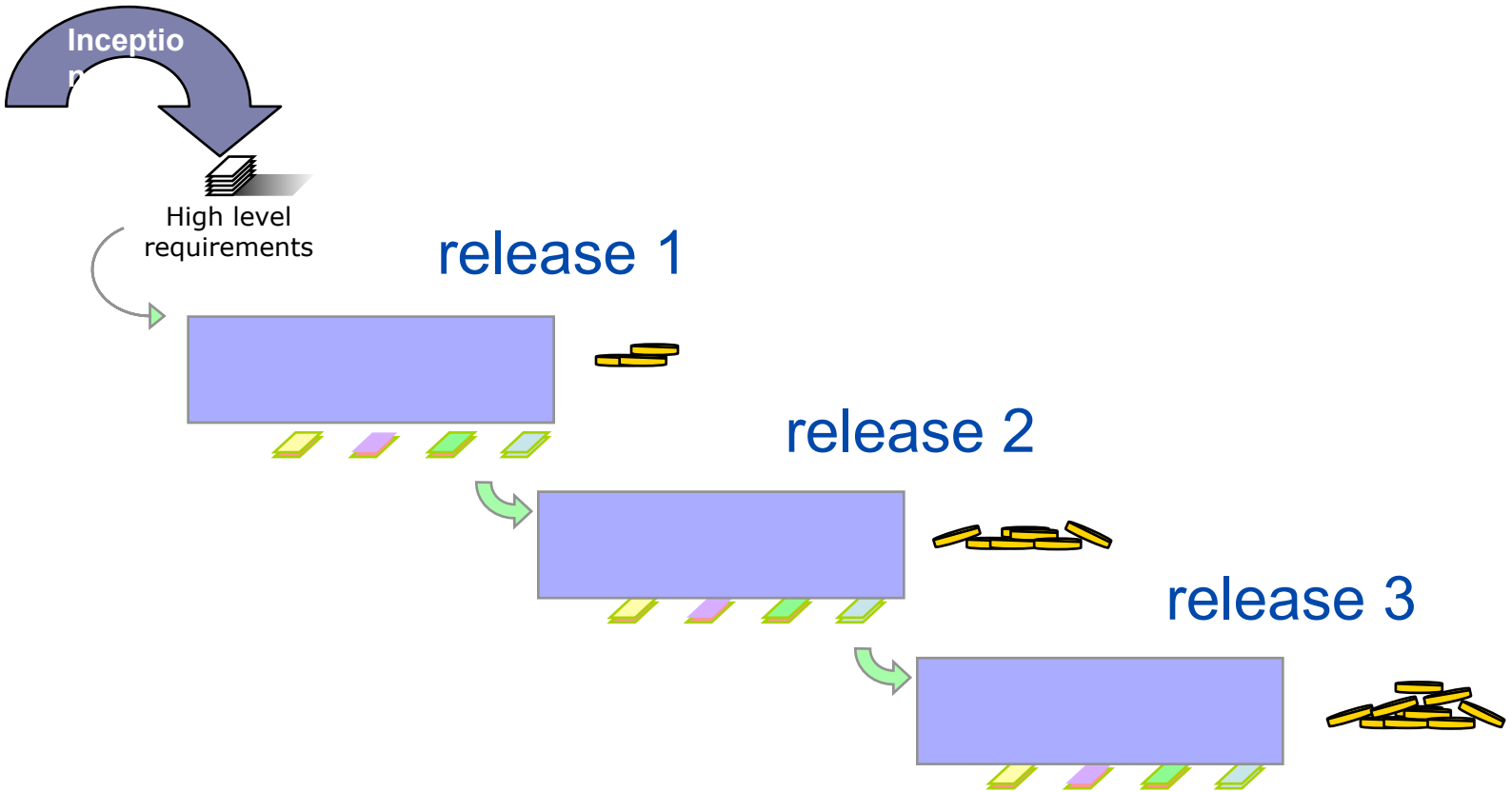
27

Agile development is an iterative and incremental process



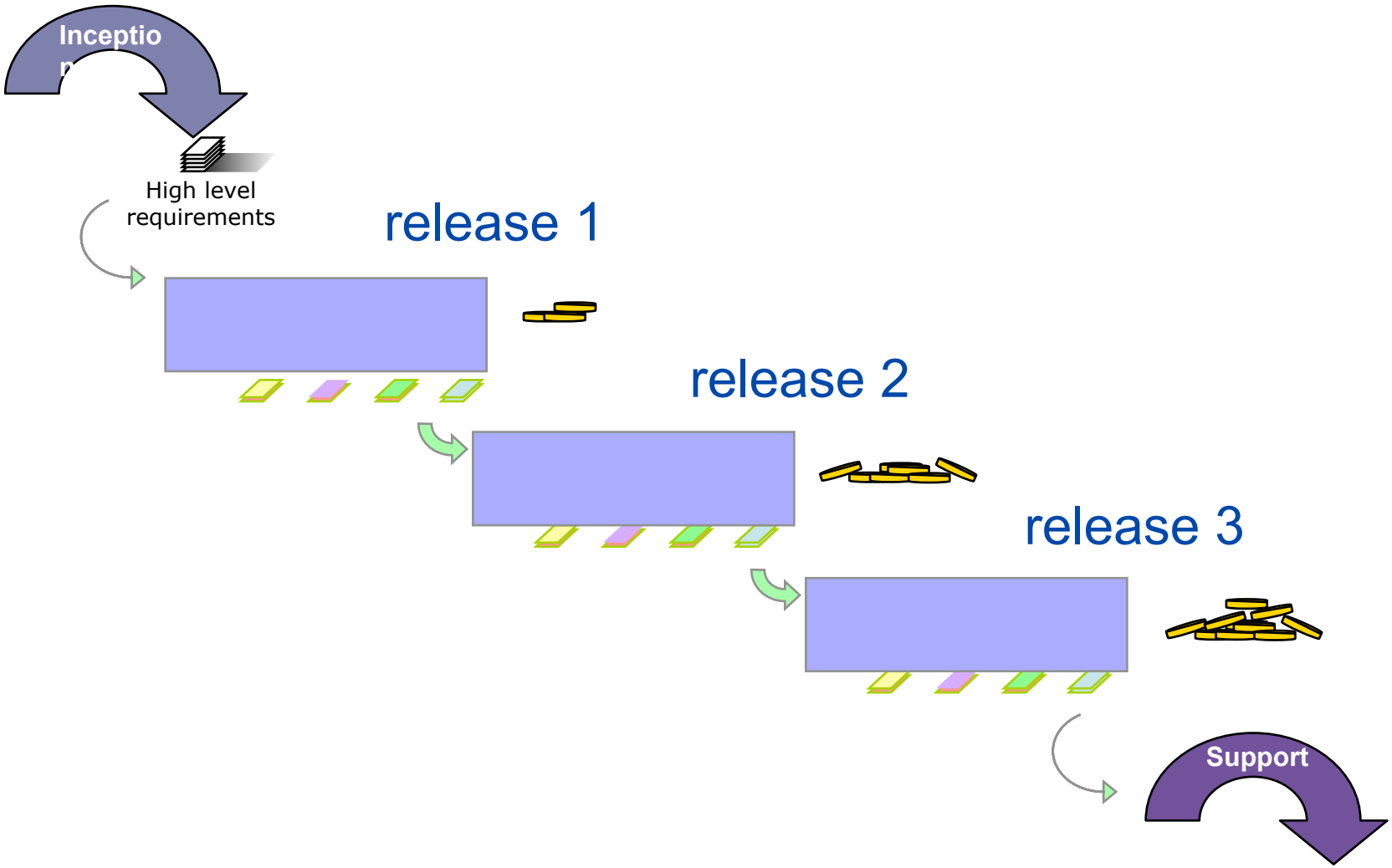
27

Agile development is an iterative and incremental process



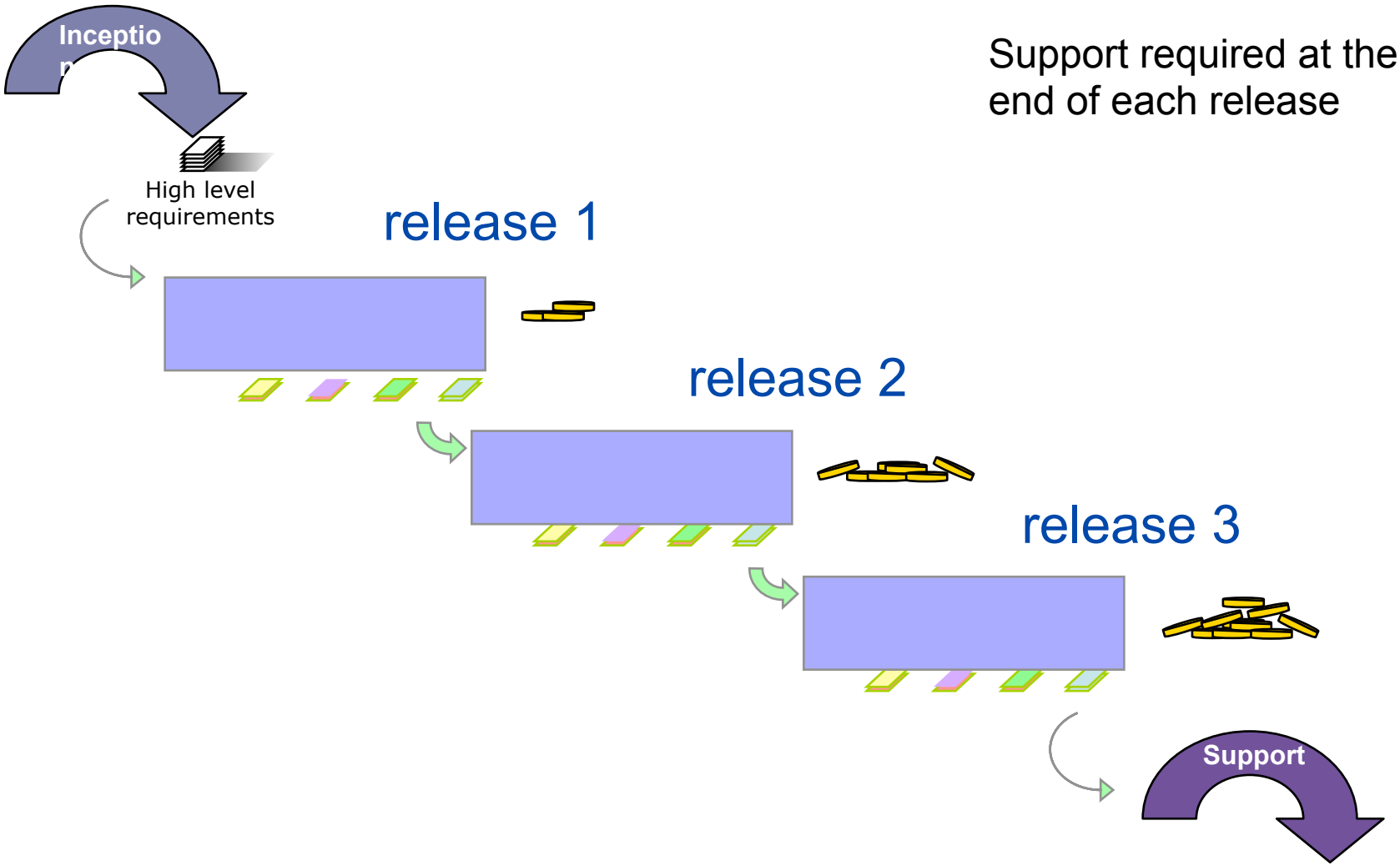
27

Agile development is an iterative and incremental process



27

Agile development is an iterative and incremental process



Support required at the end of each release

27

Agenda for this session

- The Story of Software Development
- Lean Thinking
- Agile Values and Principles
- Agile Process
- **Agile Practices**
- Summary/Review
- Questions/Close

28

Agile Practices

Pair Programming

IKO

Retrospectives

Story Wall

Daily Stand-ups

Velocity Metrics

Iterations

User Stories

Sustainable Pace

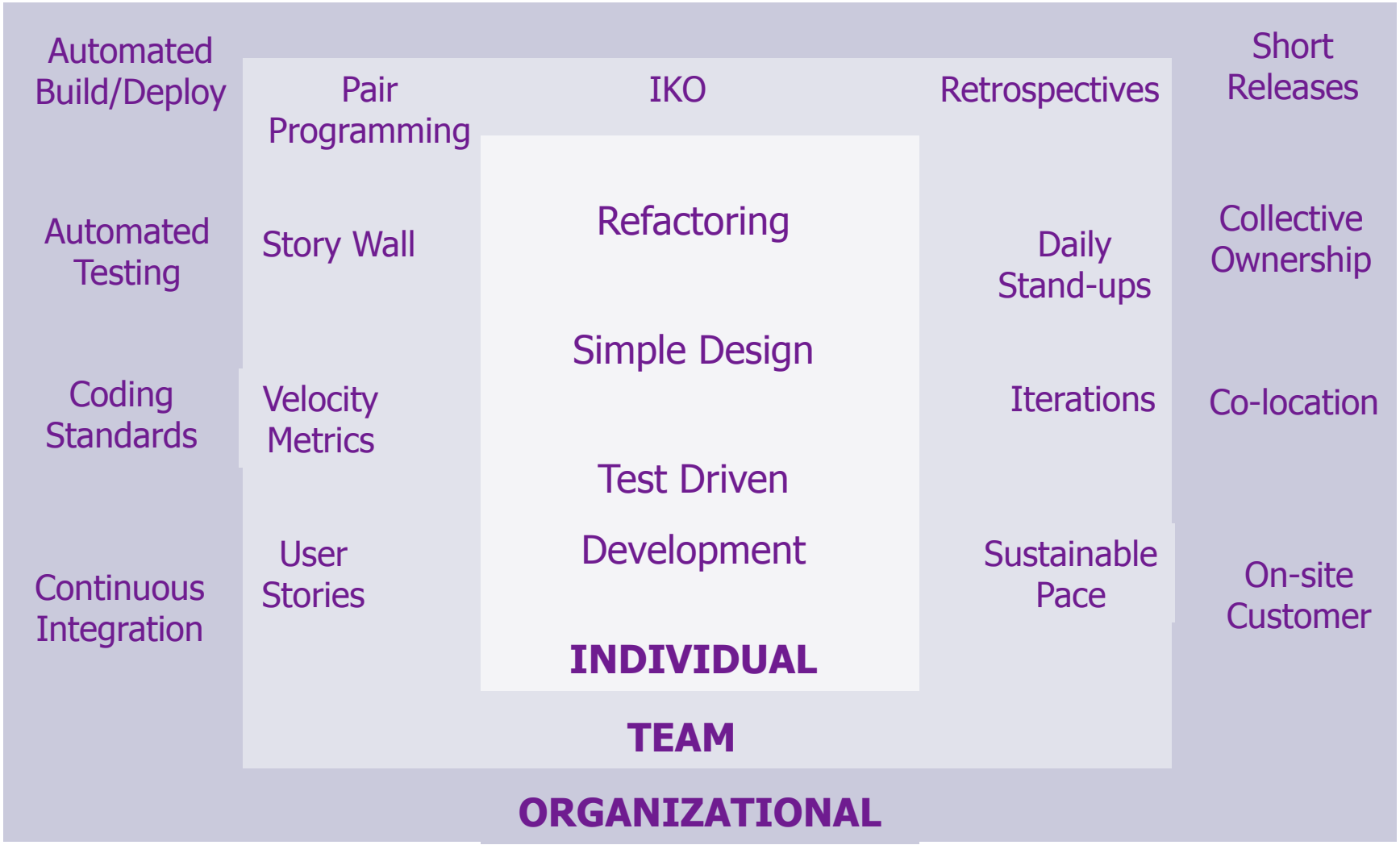
TEAM

29

Agile Practices

29

Agile Practices



Agenda for this session

- The Story of Software Development
- Lean Thinking
- Agile Values and Principles
- Agile Process
- Agile Practices
- **Summary/Review**
- Questions/Close

30

Summary

31

Summary

Use of Agile methodologies

- Helps handle changing requirements & priorities
- Lowers cost of change
- Provides better visibility into project progress
- Reduces risk
- Maximizes return on investment (business value prioritized)
- Encourages higher quality, simpler code
- Delivers business value early & often

Summary

Use of Agile methodologies

- Helps handle changing requirements & priorities
- Lowers cost of change
- Provides better visibility into project progress
- Reduces risk
- Maximizes return on investment (business value prioritized)
- Encourages higher quality, simpler code
- Delivers business value early & often

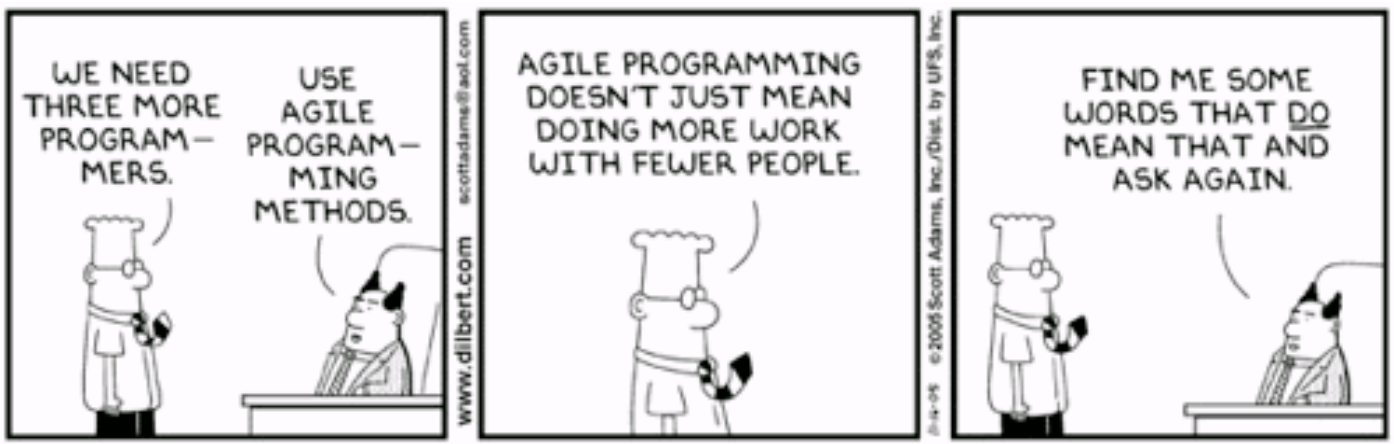
But, with this capability comes

- Constant business involvement
- A need for more discipline
- Greater emphasis on testing

31

Questions?

32



© Scott Adams, Inc./Dist. by UFS, Inc.

Agile Overview

Thanks for attending!

Balachander Swaminathan (bala@thoughtworks.com)