DAT257

Lecture 5: Technical agile

Jonas Petrén

Welcome! We will begin in a few minutes

Jonas Petrén

10 years in software testing, development and as test manager. **6+ years of Scrum Master experience.** Full-time Scrum Master since beginning of 2018. Employed by HiQ since 2012.

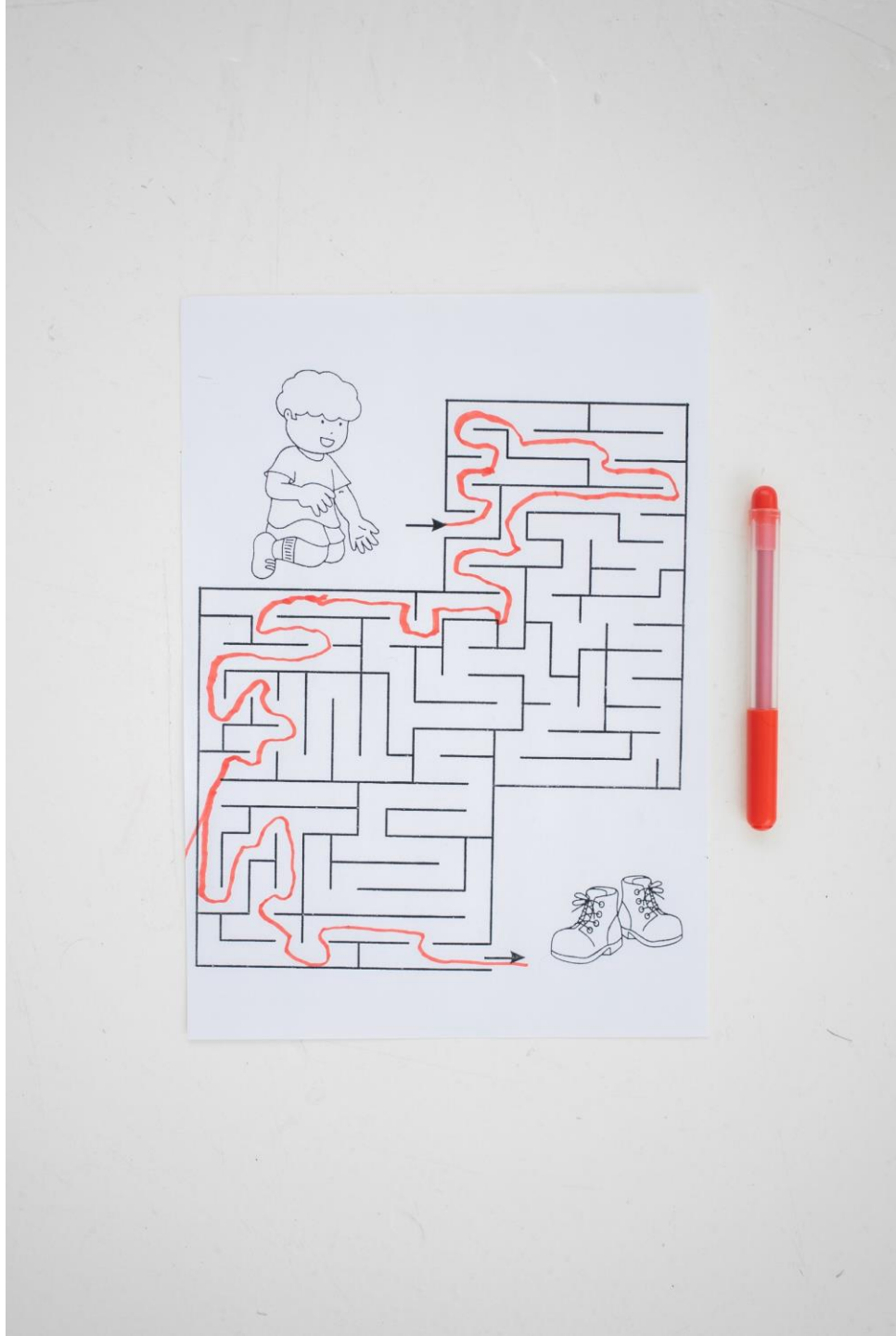
Civilingenjör Informationsteknologi, Linköping University, 2003-2008

Today: Senior Scrum Master/Agile coach

Certified **Professional Scrum Master III**

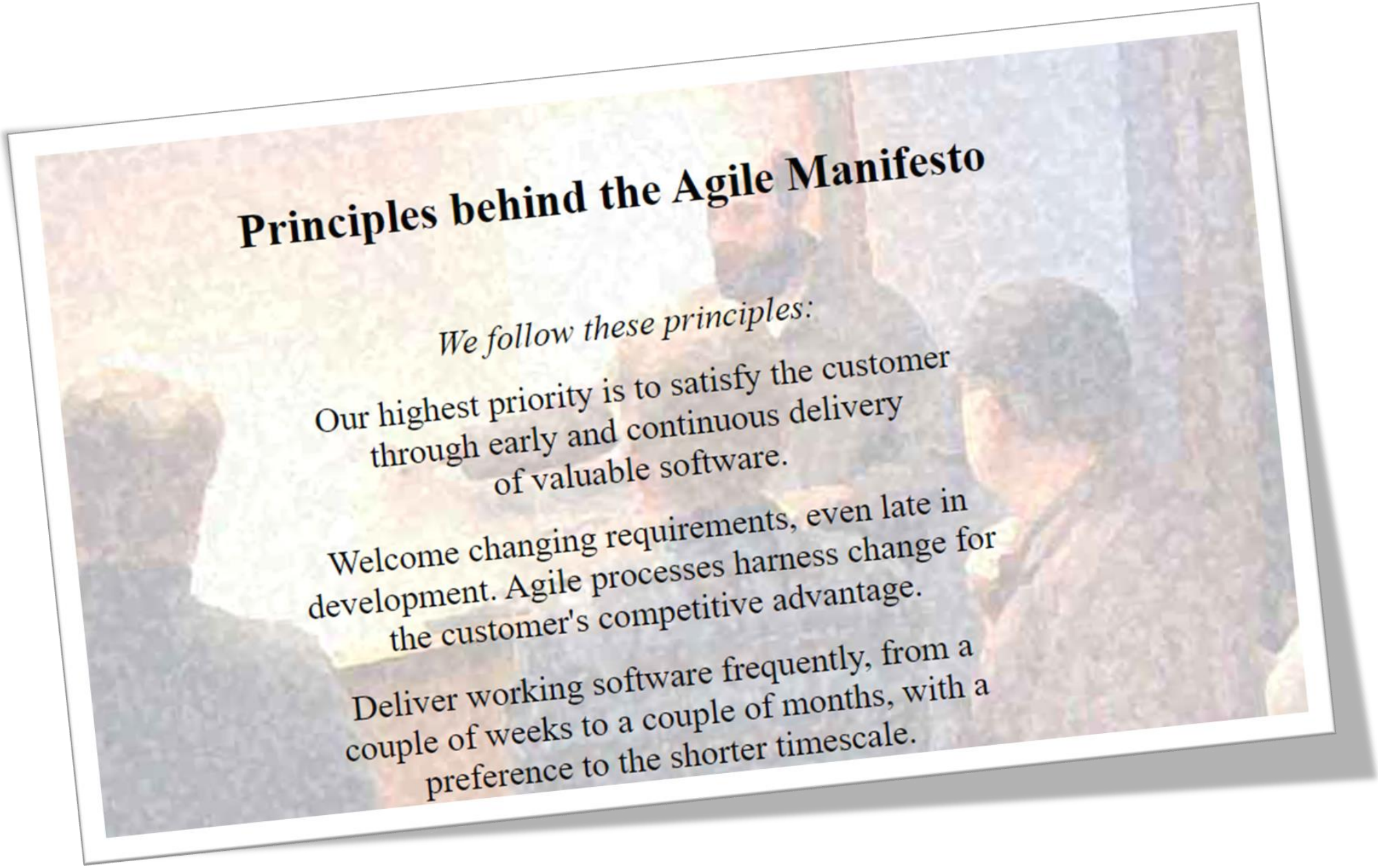
The goal for today

To get a little bit of understanding of the **technical skills** needed to build a product using agile thinking



Agenda

- Good code
- CI/CD
- DevOps
- Technical debt
- Ending



Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

[Principles behind the Agile Manifesto]

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

Principles behind the Agile Manifesto

GOOD CODE

Easy to read, understand and work with

Good code

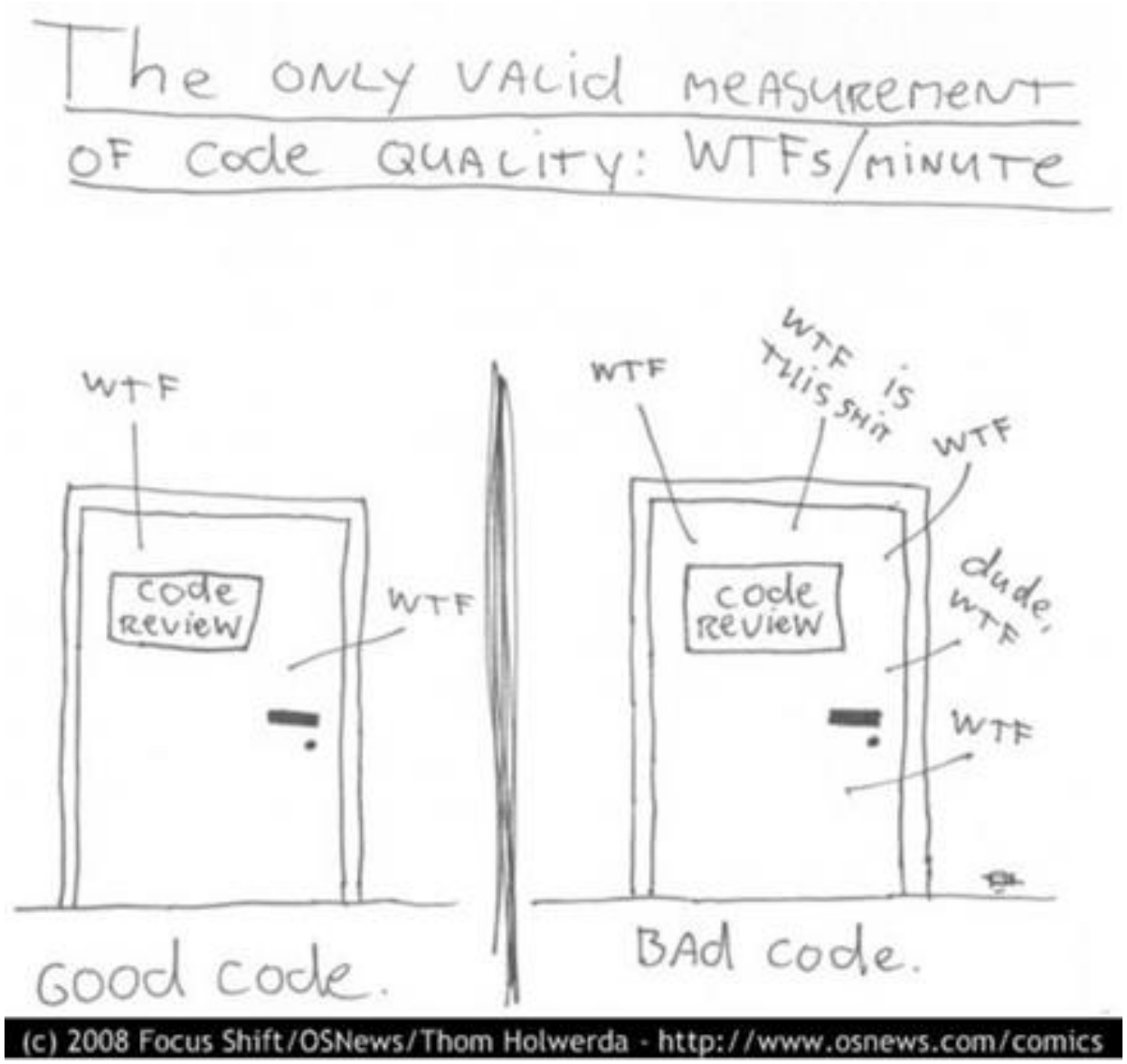
When you are working with software – 90 % of your time is reading code

Write your code so that others can understand it

Non-existing code doesn't crash. Write only the code you need

It looks like someone has programmed with their feet

Takeaway: Care for your code


The only valid measurement of code quality: WTFs/minute
WTF
code review
WTF
Good code.
WTF
WTF is this shit
WTF
code review
dude, WTF
WTF
Bad code.
(c) 2008 Focus Shift/OSNews/Thom Holwerda - http://www.osnews.com/comics

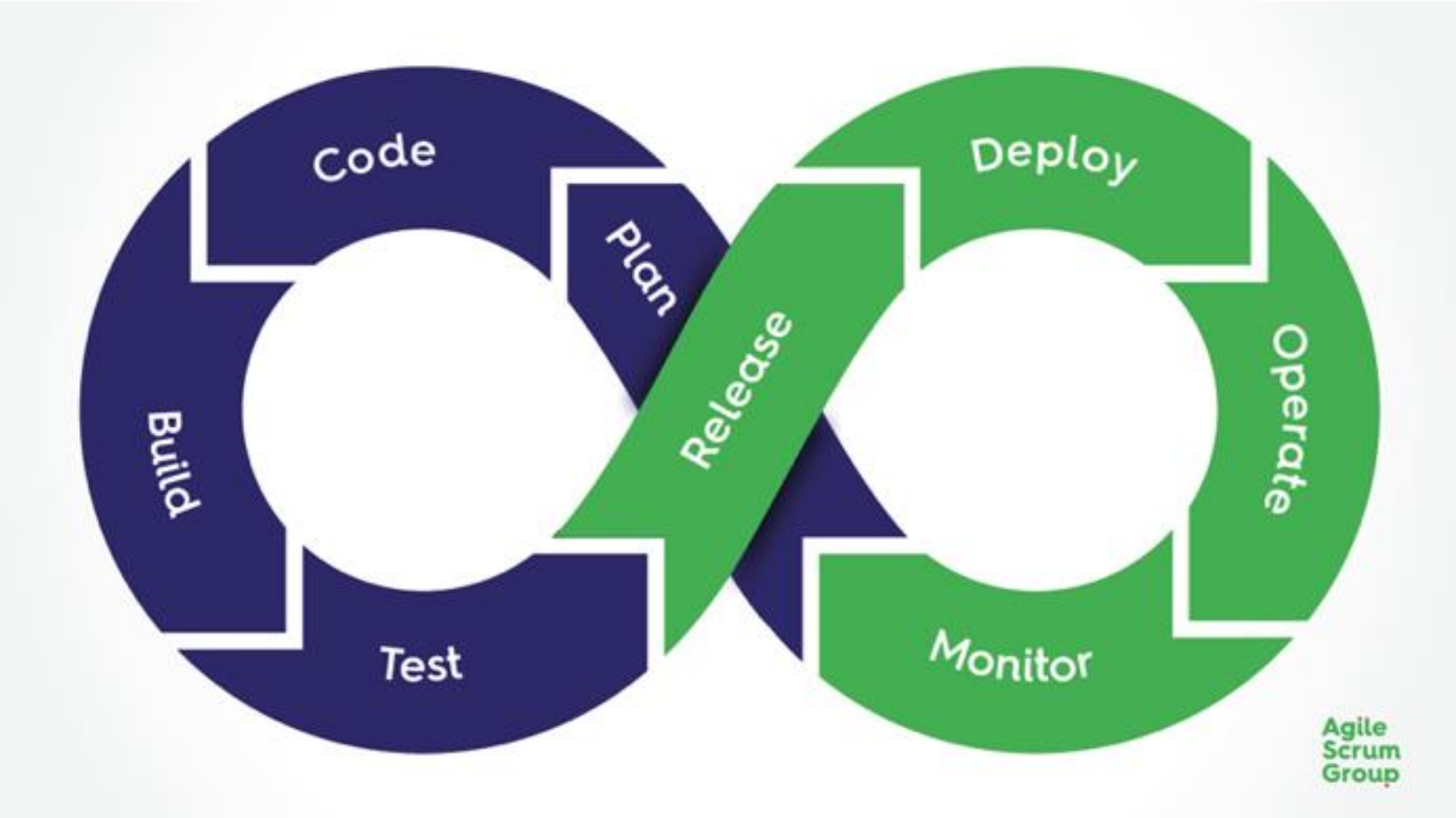
Discussion 2 minutes

Have you seen good code? If so, what made it good? How can you try to write good code in your project?

CI/CD

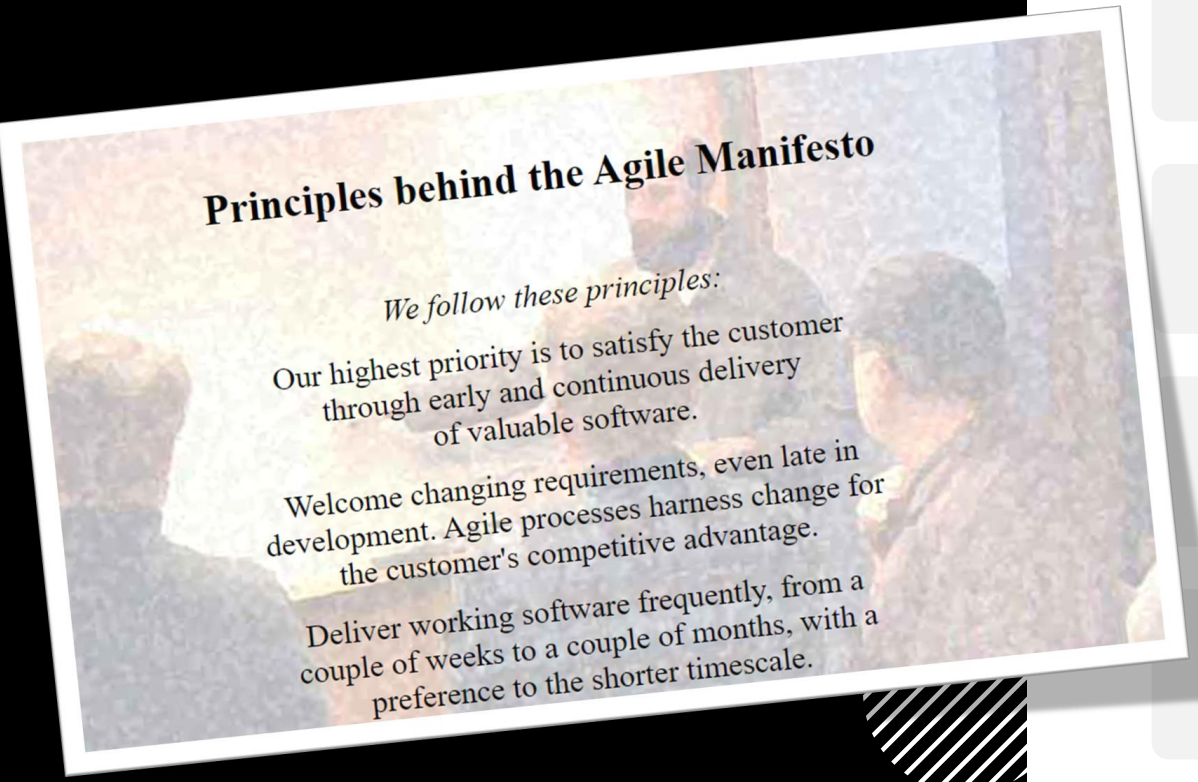
Continuous integration

Continuous delivery



[DevOps Practices in SAFe | Agile Scrum Group]

Why CI/CD?



- Early Detection of Bugs
- Enables Continuous Deployment
- Emphasize Test Automation
- Helps in determining Code Quality / Code Breaks
- Faster Development Cycle
- Improved Quality of the Product

DEVOPS

Integrating development and operations

What is DevOps?

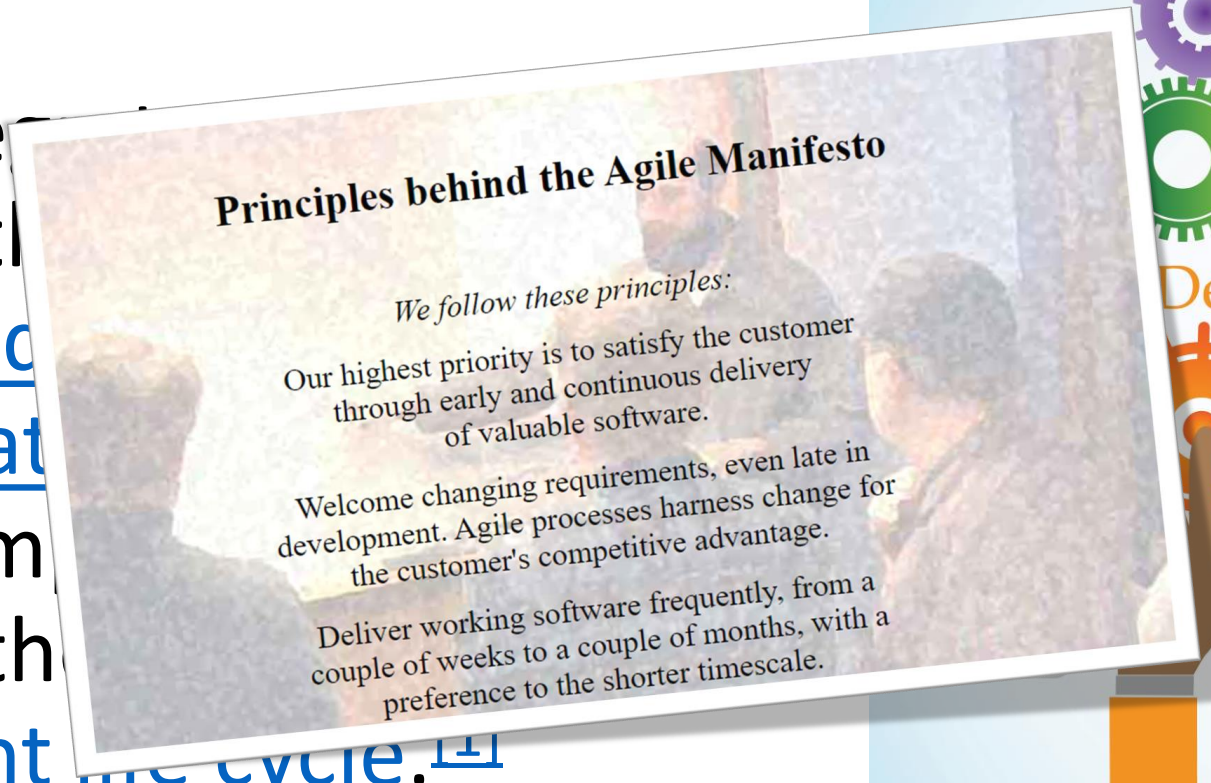
DevOps integrates and automates the work of software development (*Dev*) and IT operations (*Ops*) as a means for improving and shortening the systems development life cycle.[1]

DevOps - Wikipedia

Devops-1.jpg (3684×3541) (sazanconsulting.com)

What is DevOps?

DevOps inte
automates t
of software d
and IT operat
means for im
shortening th
development life cycle.[1]

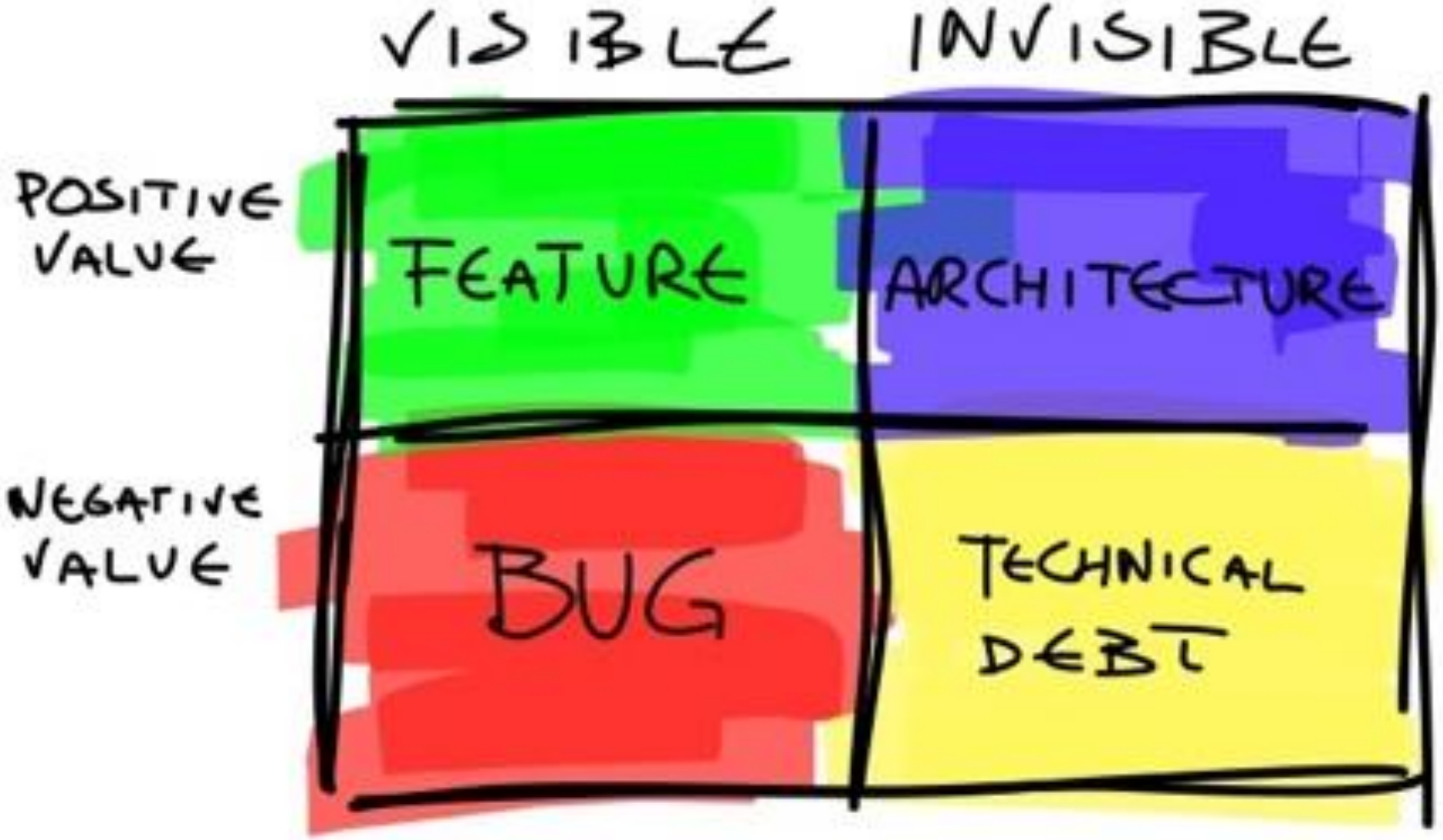


DevOps - Wikipedia

Devops-1.jpg (3684×3541) (sazanconsulting.com)

TECHNICAL DEBT

Try to keep it under control!



Machine Learning and Technical Debt with D. Sculley - Software Engineering Daily

Causes and effect

Technical debt may stem from bad design, bad architecture, time pressure, bad programming etc.

It leads to **false assumptions** about the current state of the system.

Causes and effect

Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

stem
d
ressure,
c.

assumptions about the current state of the system.

How to improve the situation

Technical debt is like the credit on your house. To reduce your debts, you must

1) **Stop taking new loans** – make no more quick-fixes or take other short cuts in the code

2) **Pay your rent** – this is the time you spend on fighting the bad code base

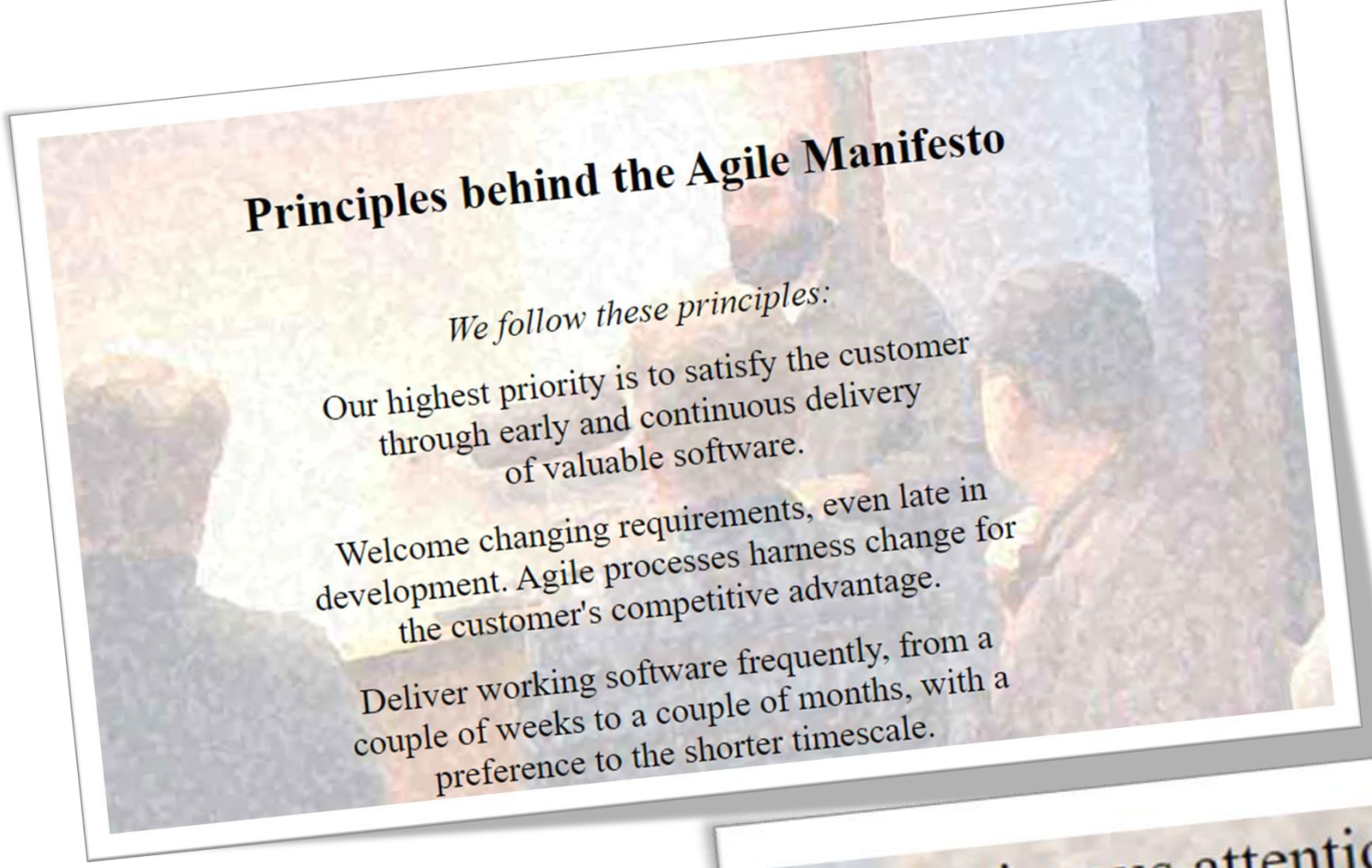
3) **Pay off your loan** – pay back your debts by refactoring the code

Key takeaways

Care for your code

Keep your technical debt under control

Technical skills are keys to a successful, long-lived product



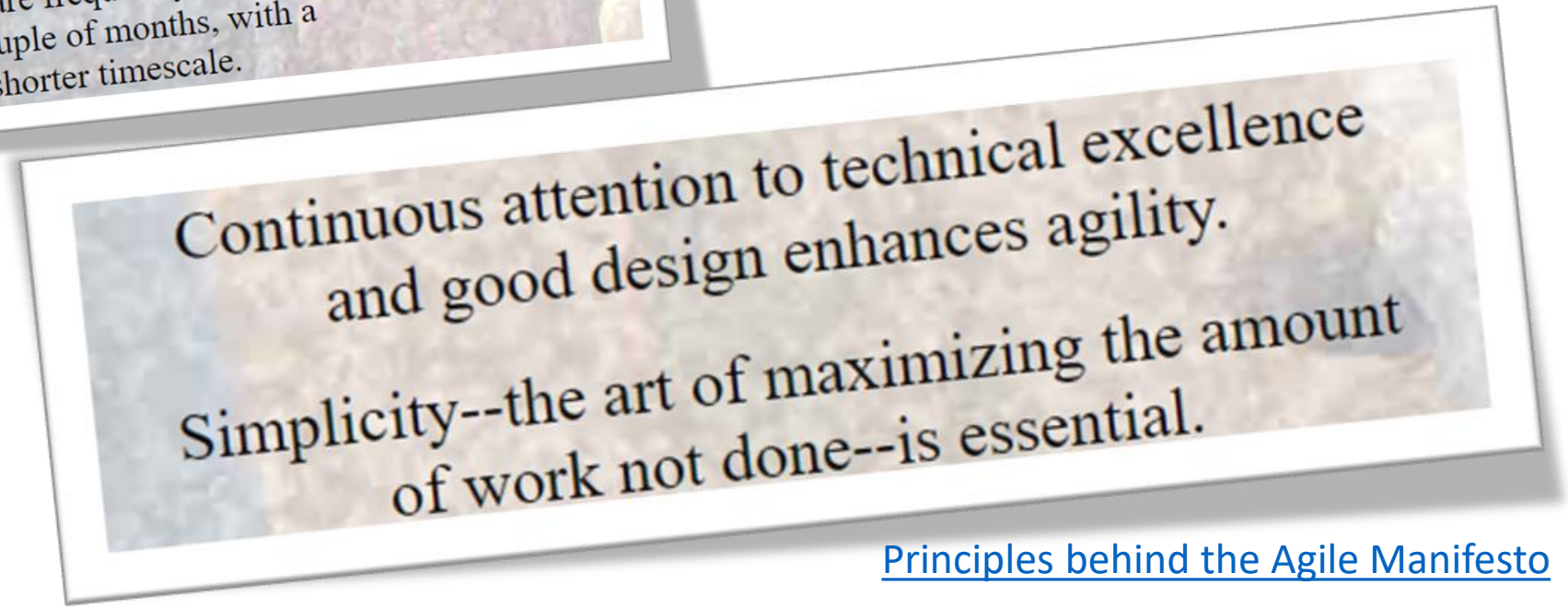
Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.



Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

[Principles behind the Agile Manifesto]


Thea Schukken
www.beeldinwerking.nl