

Welcome to the interactive presentation:

Why Scrum Projects Fail

Your hosts are:

Joseph Pelrine



Jiri Lundak



**A
BIG
FAT
FAILURE**

(Or more accurately, a tall,
really, really skinny one)

“Failure in general refers to the state or condition of not meeting a desirable or intended objective.”

- *Wikipedia*

What does failure mean in an Scrum context?

Discuss in pairs (5 Min.):

- How do you identify failure in an Scrum project?
- What failure criteria would you apply?

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

- *The Agile Manifesto's Principles*

Is this your customer?



When was your last delivery?



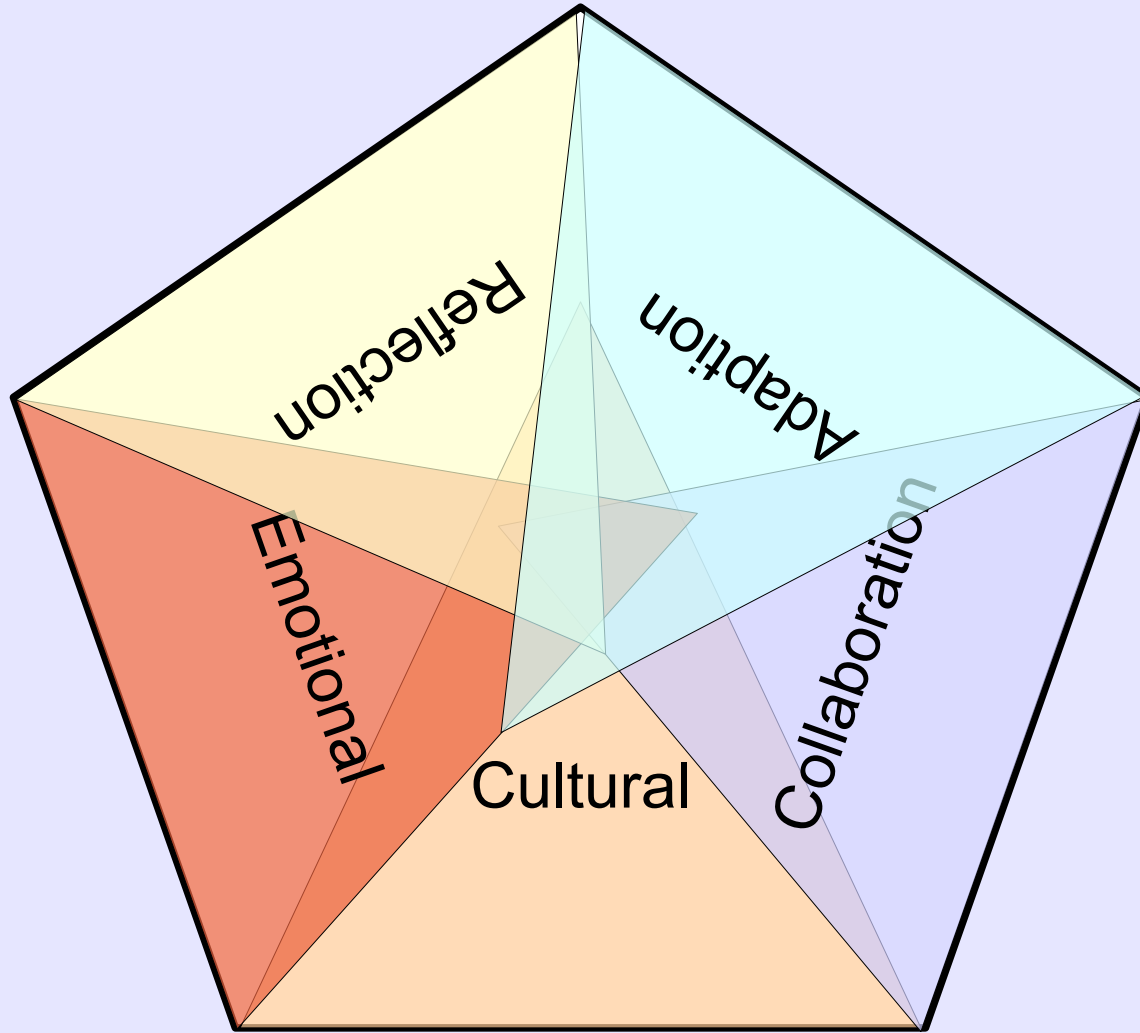
Valuable for a test drive?



“Our biggest failure is
an ***unsatisfied customer***
because of ***late or sporadic***
delivery of ***valueless artefacts.***”

- *The Failed Agile Project's Characteristics*

Scrum Problem Areas

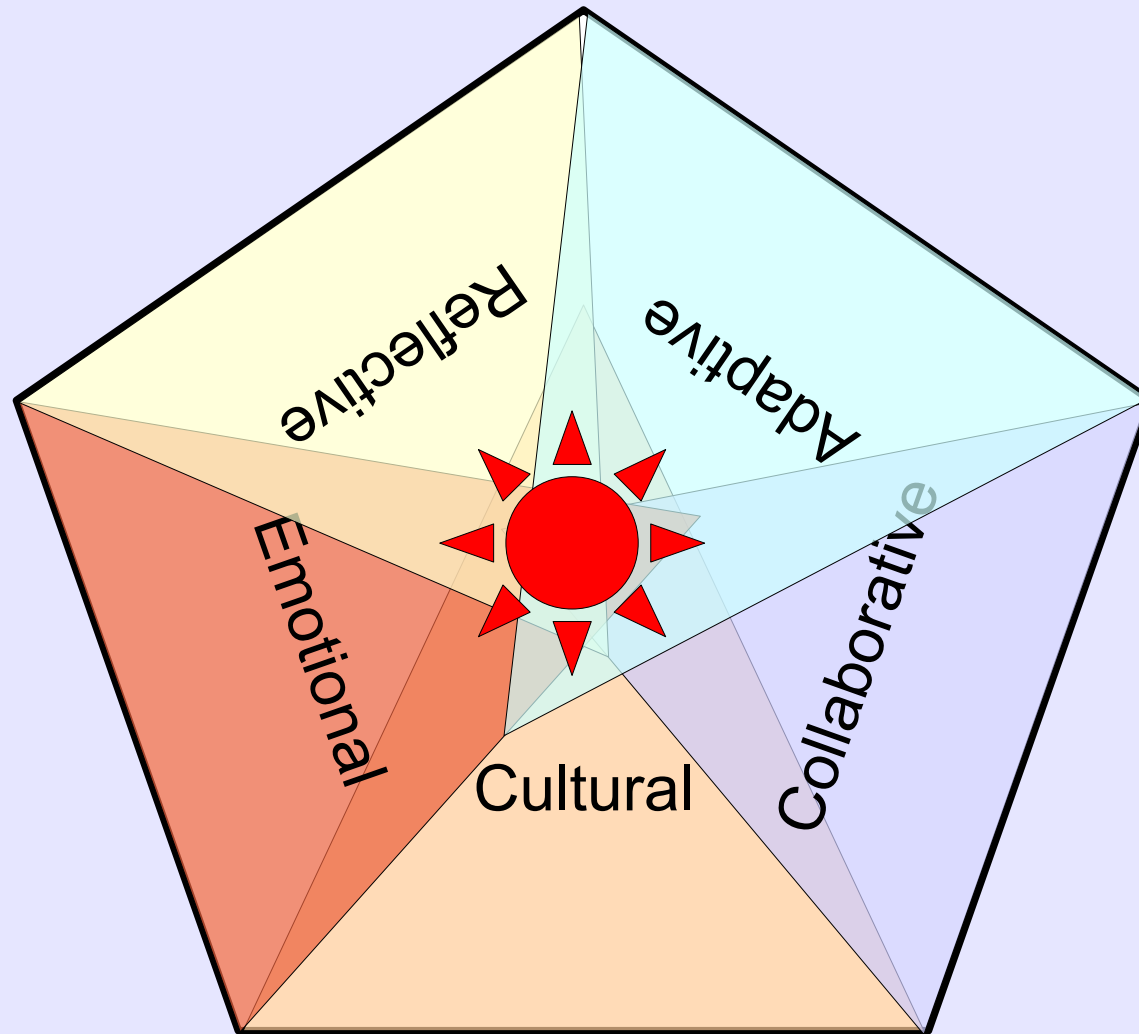


How do these Scrum problem areas influence each other?

Discuss in small groups (20 Min.):

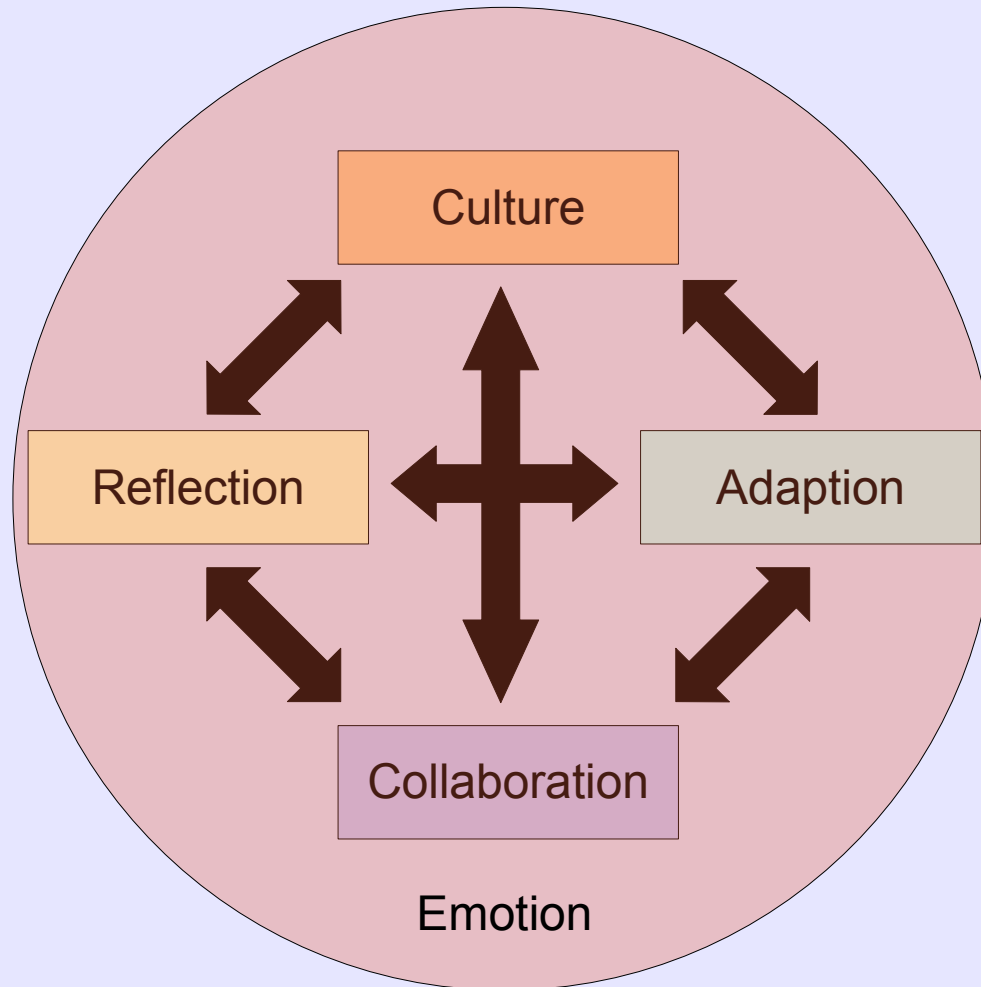
- Negative Feedback Loops
- Positive Feedback Loops

No Single Cause Makes a Scrum Project Fail!



Problem Area Dependencies

(With Emotion As a Kind of Cosmic Microwave Background Noise)



Matching symptoms with deeper causes

Note:

- Symptoms and causes are not exhaustive
- Remember the dependencies to the other problem areas

Symptoms

Individual Opposition

Upheaval

No Discipline

Subversive Behaviour

Apathy

Emotional

Fear

Influence

Antipathy (Preference)

Superiority

Ignorance

Position

Comfort

Indifference

Root Causes

Symptoms

Micromanagement

Mini-Waterfalls

Finger-Pointing

Detailed Reporting

Revoking Team-Decisions

ScrumMaster = Accountable for the Team

Cultural

Heroism

Bad Values

Command & Control

Limited Accountability

Specialization

Hierarchical Thinking

Individual Compensation

Compliance

Blaming

Root Causes

Symptoms

Process at Sprint n Same as at Sprint 1

No Known Sprint Velocity

No Hands-on Customer Demo

No Code-Reviews

Stand-up Monotonous

Tests Not Run Before Check-In

Reflection

Misleading Metrics

No Group-Reflection

Missing Self-Reflection

Missing Commitment

Non-Learning

Wrong Value Definition

Comfort Zone

Root Causes

Retrospectives Without Actions

No Time For Change

Doing Things That Do Not Work (Again)

No Refactoring

Sprint Review Without Consequences

Still No Tests

10 Ways to Do Things in 10 Days

Adaption

Ineffective ScrumMaster

Imposed Process

Cookie Cutter Process

Group Pressure

Invisible Product Owner

No Adaption

No Empowerment

Too Frequent Changes

Symptoms

Task Handover

Many Loose Ends

Only BA Talks to Customer

No Team Planning

Not Slowing Down for Teammates

Deciding For the Customer

Check-In Race

Collaboration

Segregation

Hard-coded Communication Paths

Push-System

No Shared Responsibility

Turf Wars

Separation

Politics

Root Causes

How can we identify root causes?

Principles:

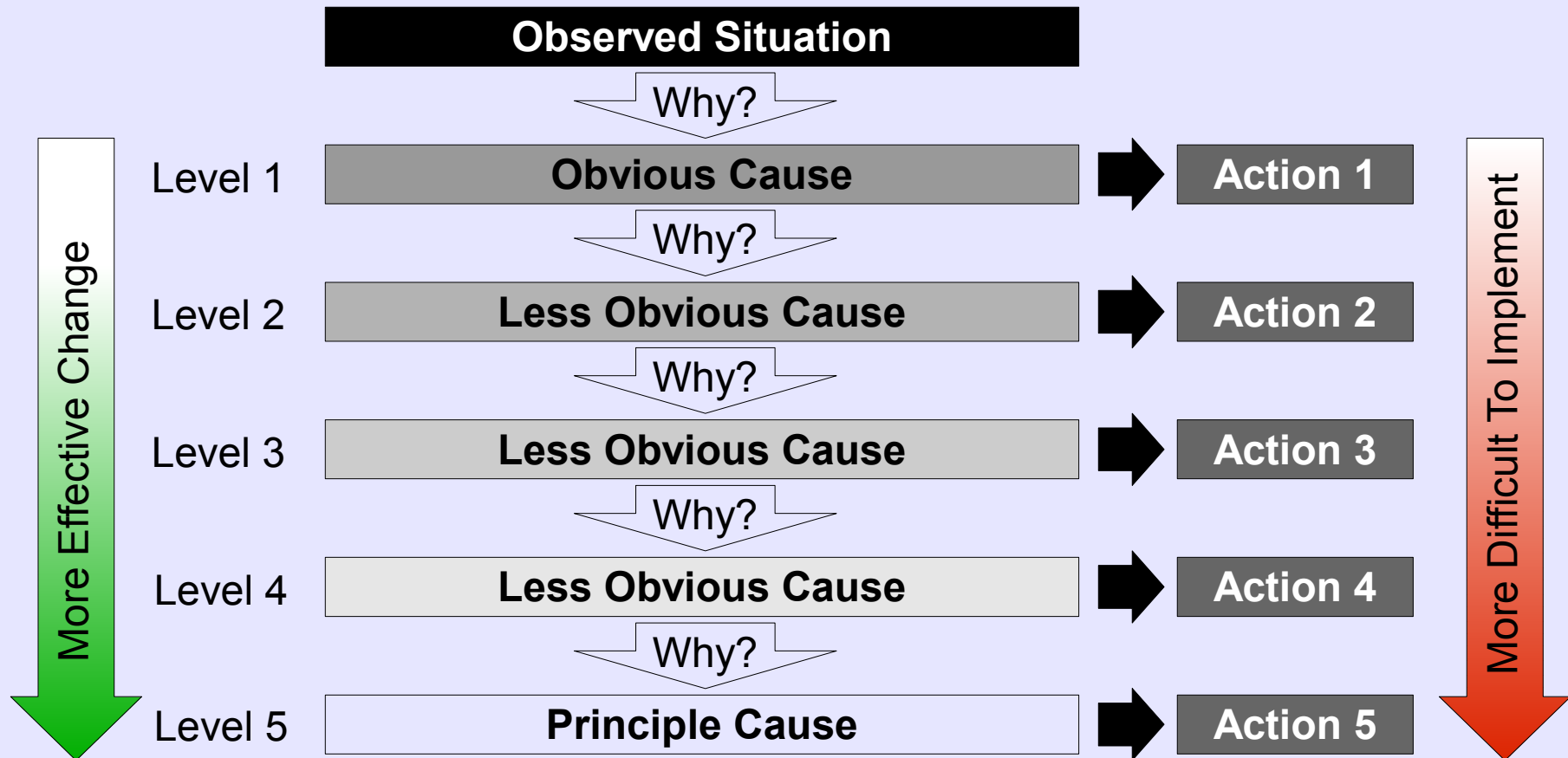
- Retrospective Coherence

Techniques:

- Toyotas' "Five Why"-Method
- Ishikawa-/Fishbone Diagram

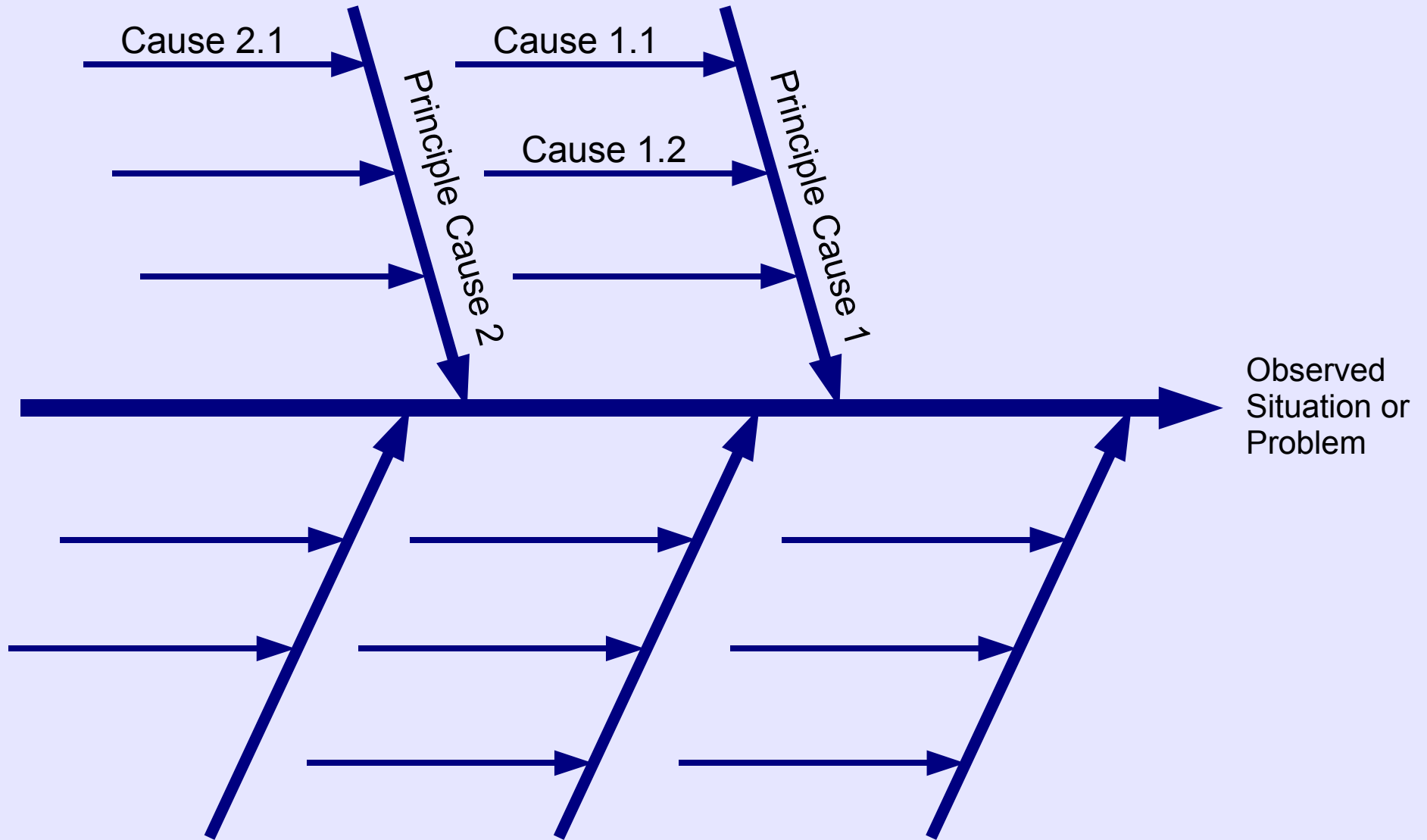
Searching for Root Causes

(Using Toyota's "5 Why"-Method)



Searching for Root Causes

(Using the Ishikawa-/Fishbone-Diagram)



A Sample Failure Story:

“A team does not deliver potentially shippable code at Sprint's end. Using a task board you see lots of tasks in the 'To Be Tested' column.

Nonetheless they show the system to the customer and the customer is still satisfied with the result. The team continues this way for five Sprints.

The system seems feature complete and a real production test date is set.

The system fails the production test and it takes the team another three Sprints to fix all bugs in the System, going over budget and having not a good reputation with the customer.

But in the end the troubled project is seen as a success anyway.”

Discuss:

- **Was this project a real Scrum success story or a failure?**
- **What are the symptoms of deeper problems?**
- **What are the root causes for the problems we could observe?**

Applying root cause analysis for the sample failure story

Techniques:

- Toyotas' "Five Why"-Method
- Ishikawa-/Fishbone Diagram

Tell your personal failure story and do a root cause analysis

Do in small groups (30 Min.):

- Tell and write up at least two failure stories
- Fill in the story meta-data
(according to the Agile Narratives initiative)
- Find the root causes

Comparing root causes for Scrum failure

Final Discussion (15 Min.):

- Are common patterns observable?
- How do we address these problems?

Thank You!

How Scrum Projects Can Succeed...

For further information and an
updated version of this presentation visit

www.metaprog.com

www.loewenfels.ch

or send an e-mail to

jpelrine@metaprog.com

jiri.lundak@loewenfels.ch