

12. – 15.09.2010
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

V34

Selbstorganisiert ein Ziel erreichen

Analyse, Architektur und Design in agilen Software-Projekten

David Tanzer

Selbstorganisiert ein Ziel erreichen

Analyse, Architektur und Design in agilen Software-Projekten



Über den Vortragenden

Freiberuflicher SW-Entwickler
und Berater seit 2006

Certified Scrum Master

Bakk. Techn. (Johannes Kepler
Universität Linz)



Scrum – Erfahrung

Seit dem Studium an Agiler SW-Entwicklung interessiert

Entwickler in einem Scrum-Team von Sommer 2009 bis Mai
2010

Das Team war das erste Scrum-Team des Konzerns

The best architectures, requirements, and designs emerge from self-organizing teams.

Continuous attention to technical excellence and good design enhances agility.



Agenda

Begriffserklärung

Analyse

Design

Architektur

Agile Methoden

Selbstorganisation

Scrum

Zeitplan Sprint

Analyse

Design

Architektur



Agenda

Begriffserklärung

Analyse

Design

Architektur

Agile Methoden

Selbstorganisation

Scrum

Zeitplan Sprint

Analyse

Design

Architektur



Laut IEEE kann die Anforderungserhebung in Anforderungsaufnahme (requirements elicitation), Anforderungsanalyse (requirements analysis), Anforderungsspezifikation (requirements specification) und Anforderungsbewertung (requirements validation) unterteilt werden.

Verstehen was der Kunde will.

In most successful software projects, the expert developers working on that project have a shared understanding of the system design. This shared understanding is called 'architecture.'

This understanding includes how the system is divided into components and how the components interact through interfaces.

These components are usually composed of smaller components, but the architecture only includes the components and interfaces that are understood by all the developers.

Software architecture is the realization of non-functional requirements, while software design is the realization of functional requirements.

Software architecture, also described as strategic design, is an activity concerned with global design constraints, such as programming paradigms, architectural styles, component-based software engineering standards, design principles, and law-governed regularities.

Detailed design, also described as tactical design, is an activity concerned with local design constraints, such as design patterns, architectural patterns, programming idioms, and refactorings.

Agenda

Begriffserklärung

Analyse

Design

Architektur

Agile Methoden

Selbstorganisation

Scrum

Zeitplan Sprint

Analyse

Design

Architektur



Selbstorganisation kann nur passieren, wenn ...

- es einen Rahmen gibt, der die Möglichkeiten der Selbstorganisation begrenzt

- es Unterschiede innerhalb der Teammitglieder gibt

- es Austausch zwischen den Teammitgliedern gibt

Um ein gemeinsames Ziel zu erreichen, muss man ...

- wissen, wo das Ziel ist

 - ... auch wenn es sich ändern kann

- wissen, wo man gerade steht (zumindest ungefähr)

- das erste Stück des Weges sehen

Das Team ist gemeinsam für das Gesamtergebnis verantwortlich

Jeder im Team ist für Architektur verantwortlich

Und für Anforderungen

Und für Testen

Und für Entwicklung

...

Klar gibt es Spezialisten ...

... aber sie müssen auch in anderen Spezialgebieten „aushelfen“



Scrum Master

Scrum - Team

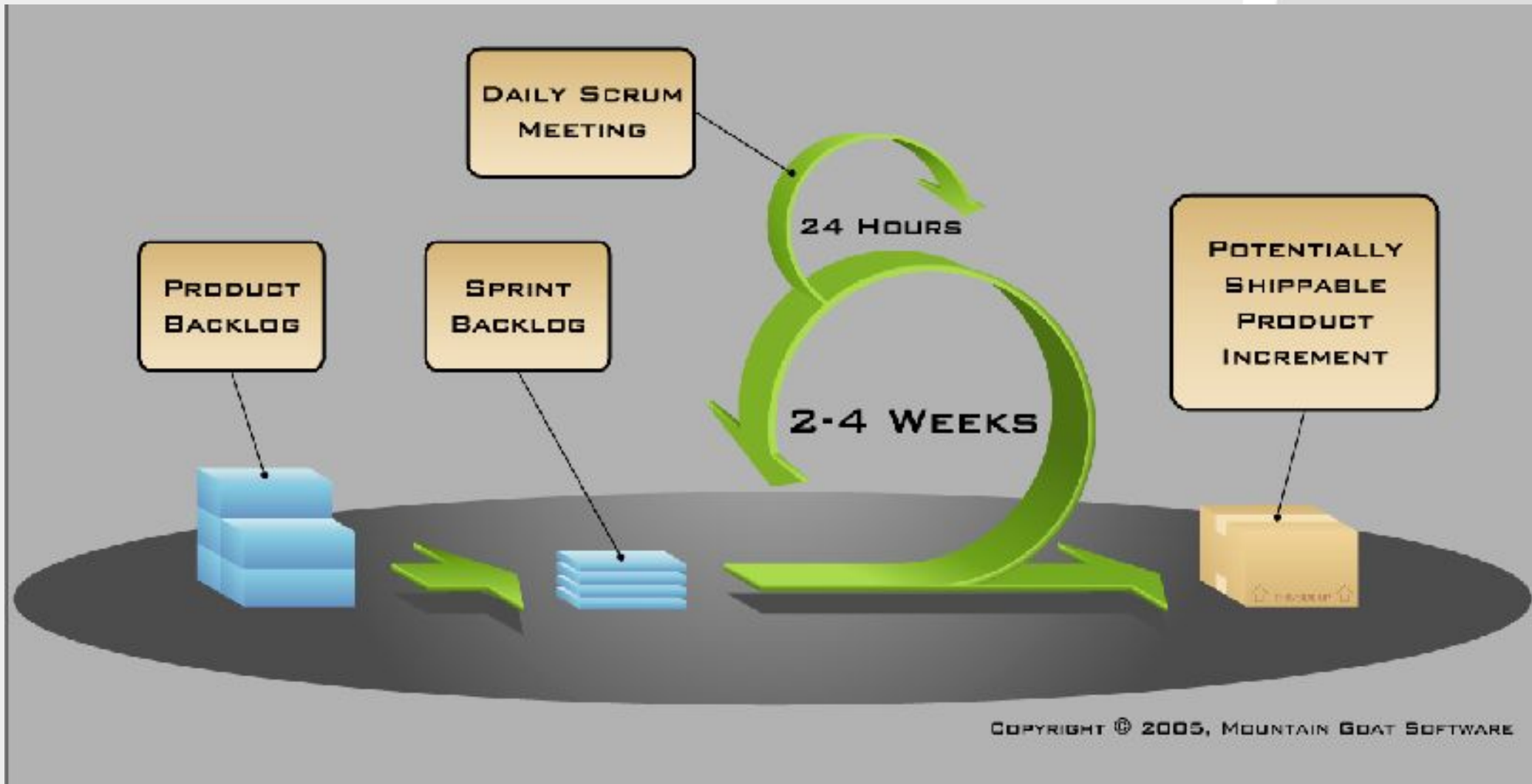


Product Owner

Entwicklerteam



Scrum (Kurzfassung)



Zeitplan

 Daily Scrum

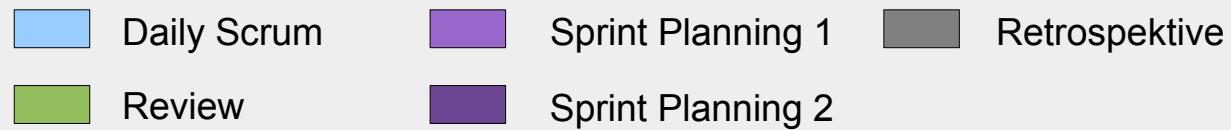
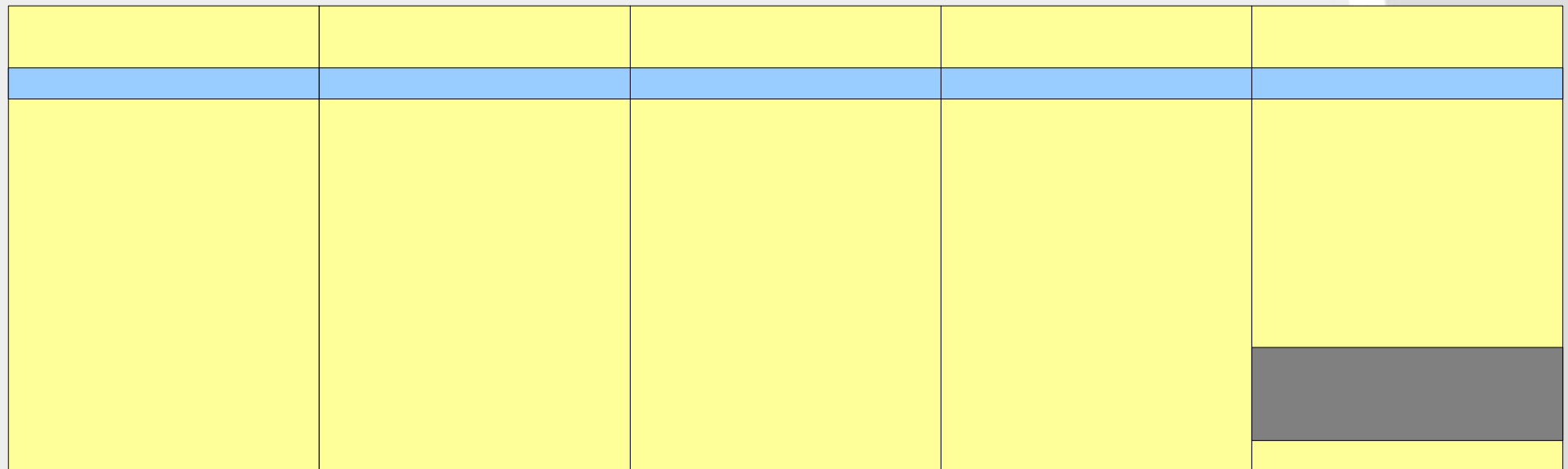
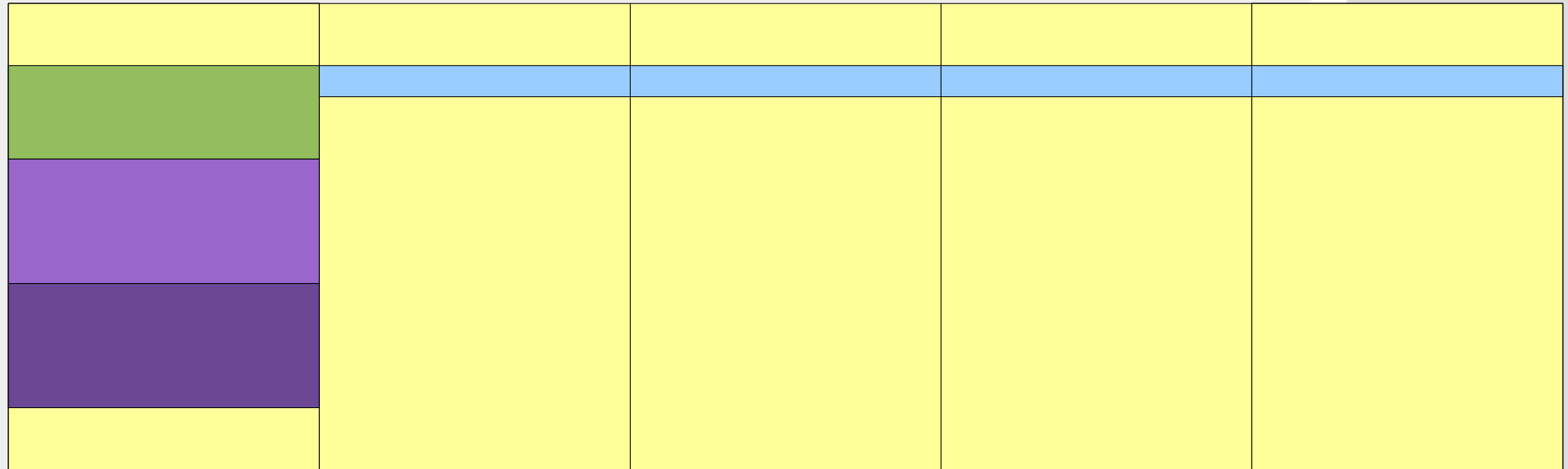
Zeitplan

[illegible]

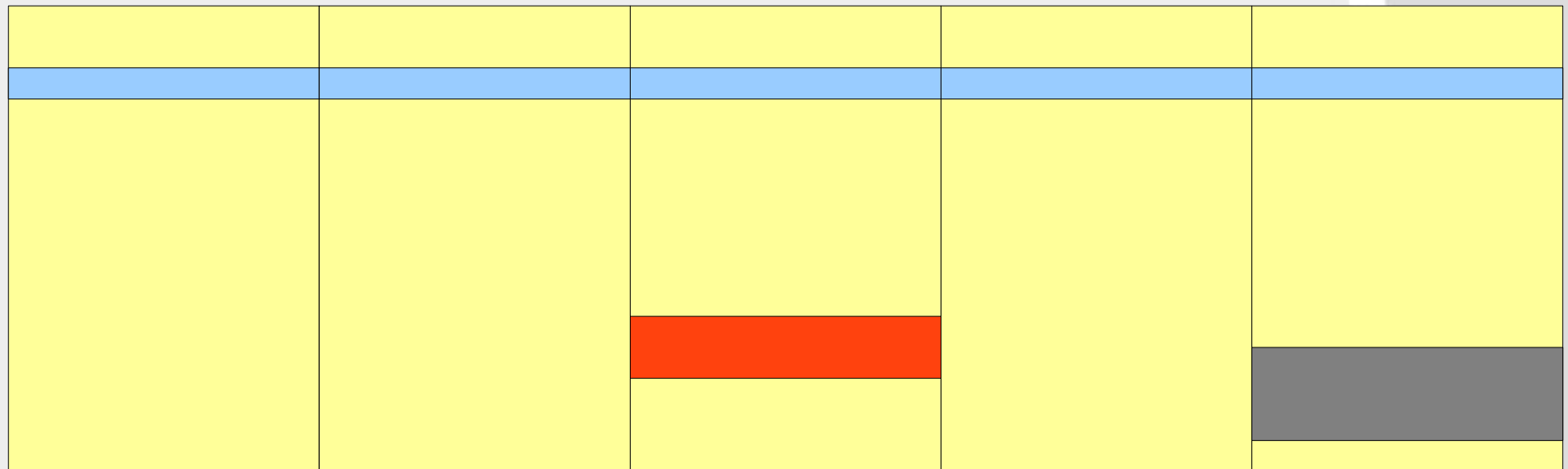
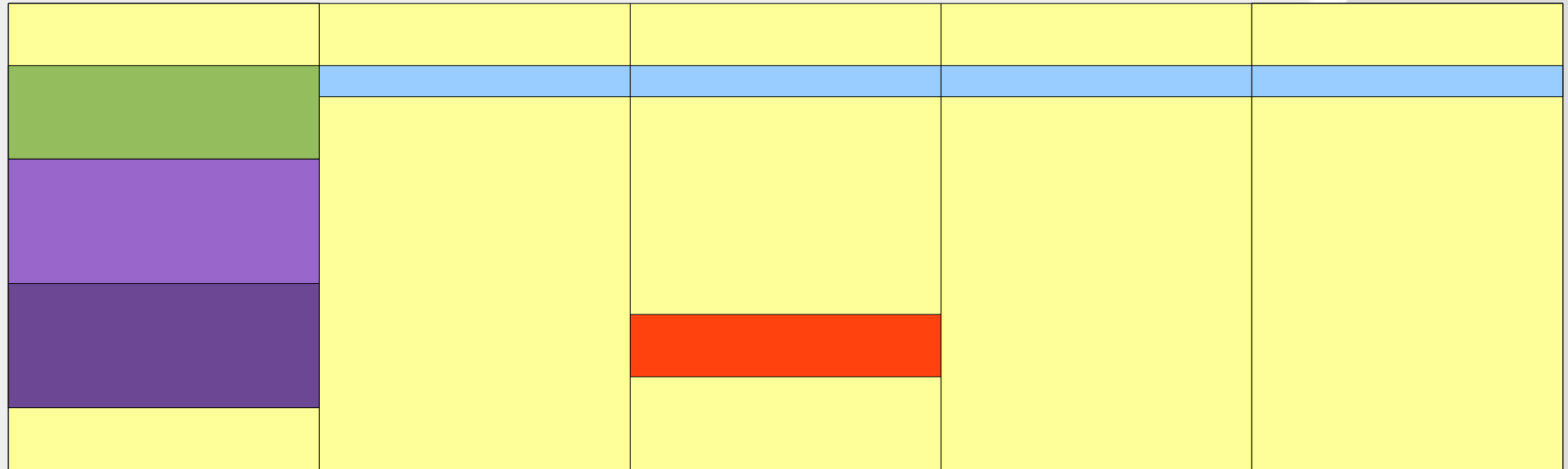
Legend for the Scrum process diagram:

- Daily Scrum (Light Blue)
- Review (Light Green)
- Sprint Planning 1 (Light Purple)
- Sprint Planning 2 (Dark Purple)

Zeitplan



Zeitplan



- | | | |
|---|---|--|
|  Daily Scrum |  Sprint Planning 1 |  Retrospektive |
|  Review |  Sprint Planning 2 |  Backlog-Pflege ("Schätzmeeting") |

Agenda

Begriffserklärung

Analyse

Design

Architektur

Agile Methoden

Selbstorganisation

Scrum

Zeitplan Sprint

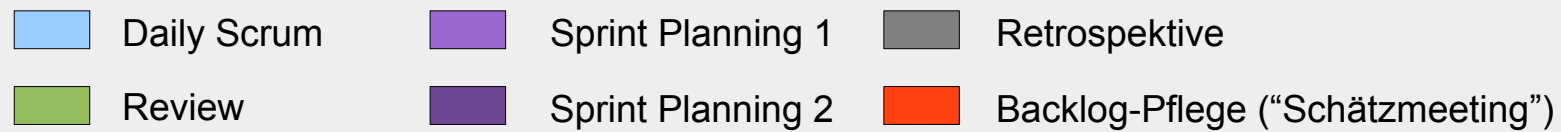
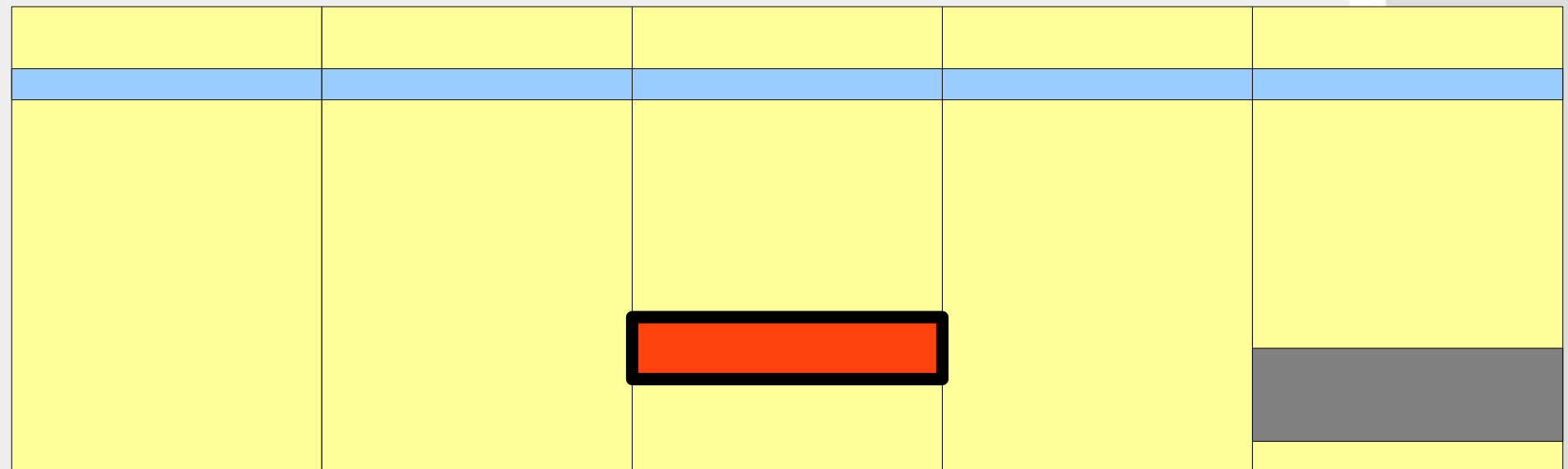
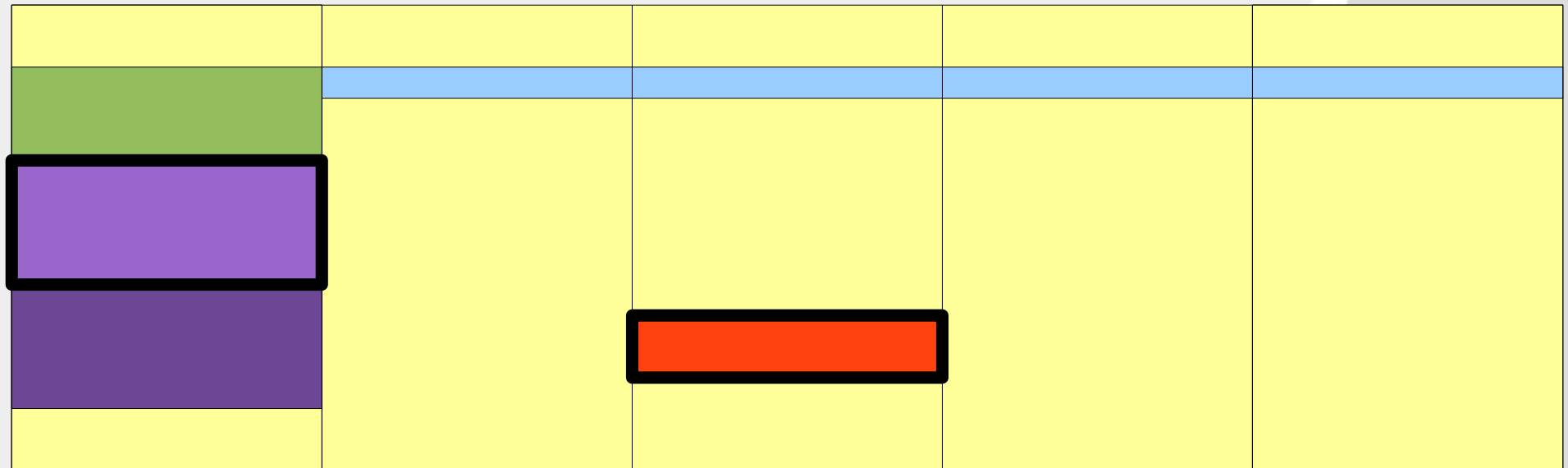
Analyse

Design

Architektur



Zeitplan



Agenda

Begriffserklärung

Analyse

Design

Architektur

Agile Methoden

Selbstorganisation

Scrum

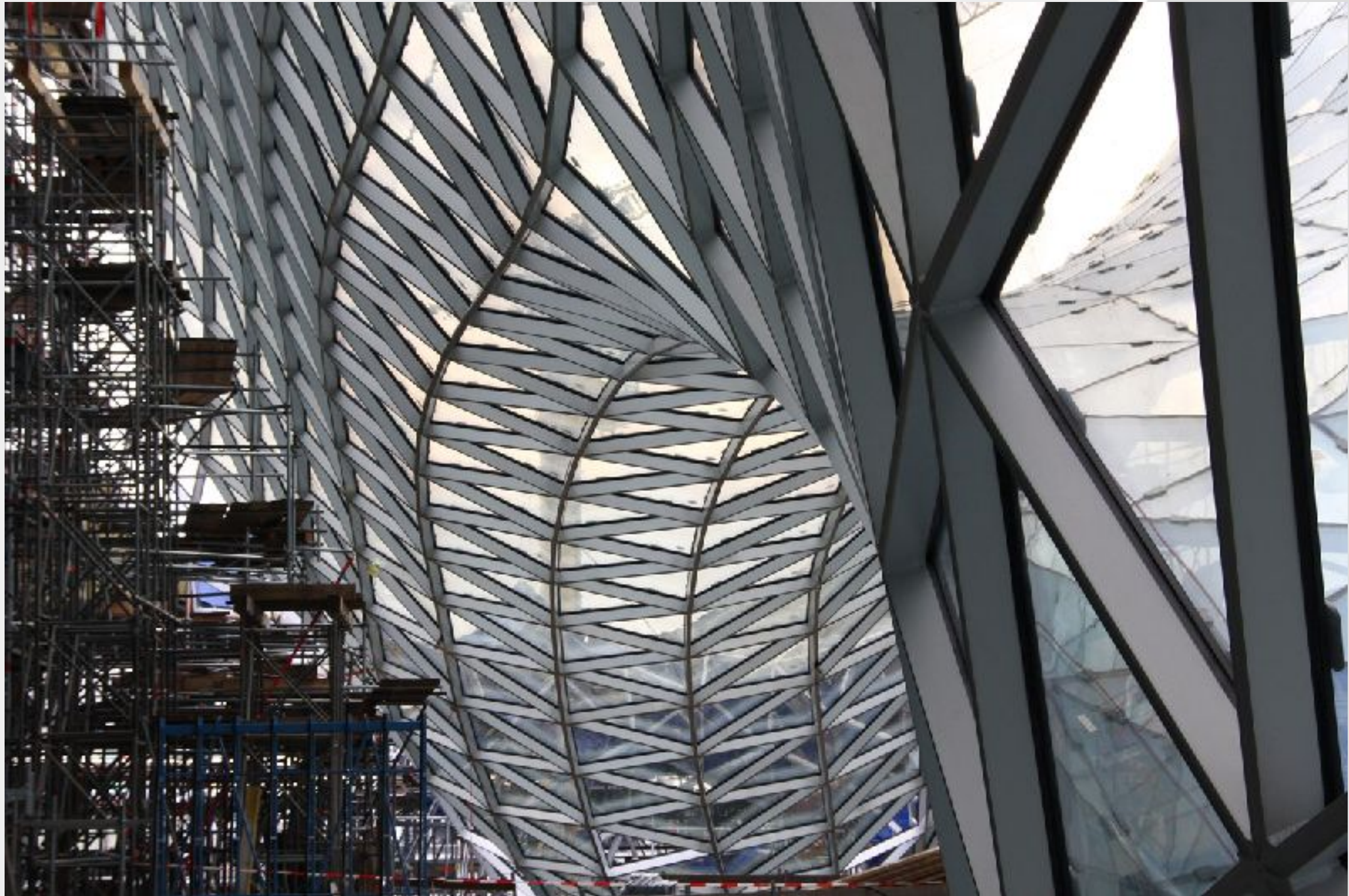
Zeitplan Sprint

Analyse

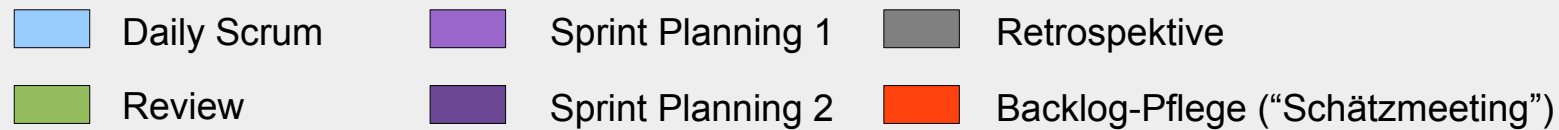
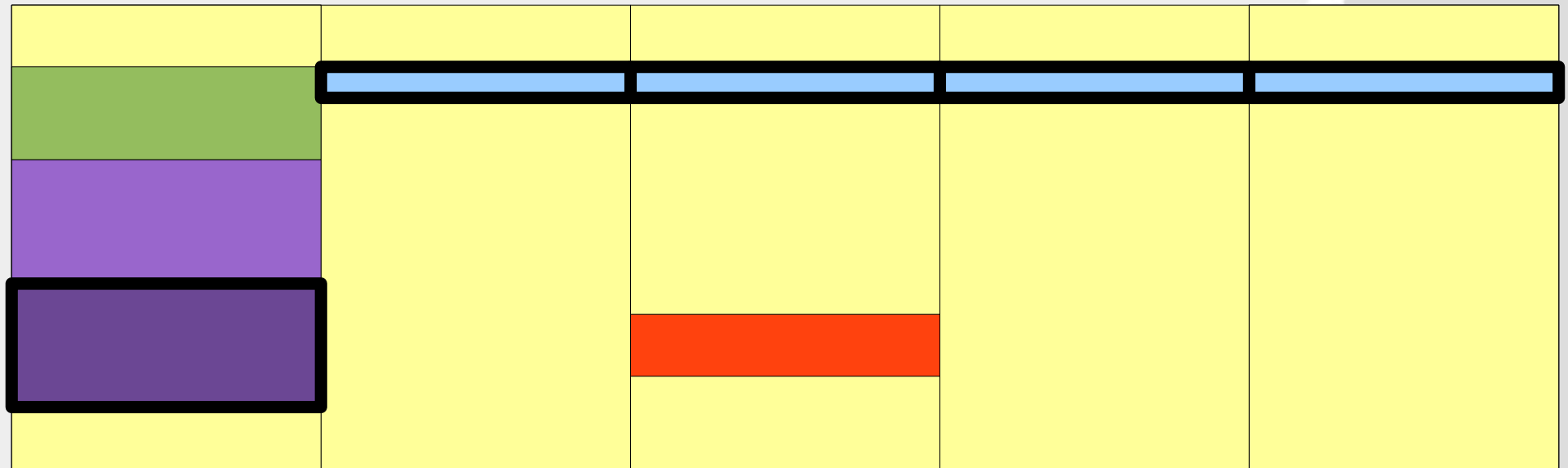
Design

Architektur





Zeitplan



Agenda

Begriffserklärung

Analyse

Architektur

Design

Agile Methoden

Selbstorganisation

Scrum

Zeitplan Sprint

Analyse

Design

Architektur





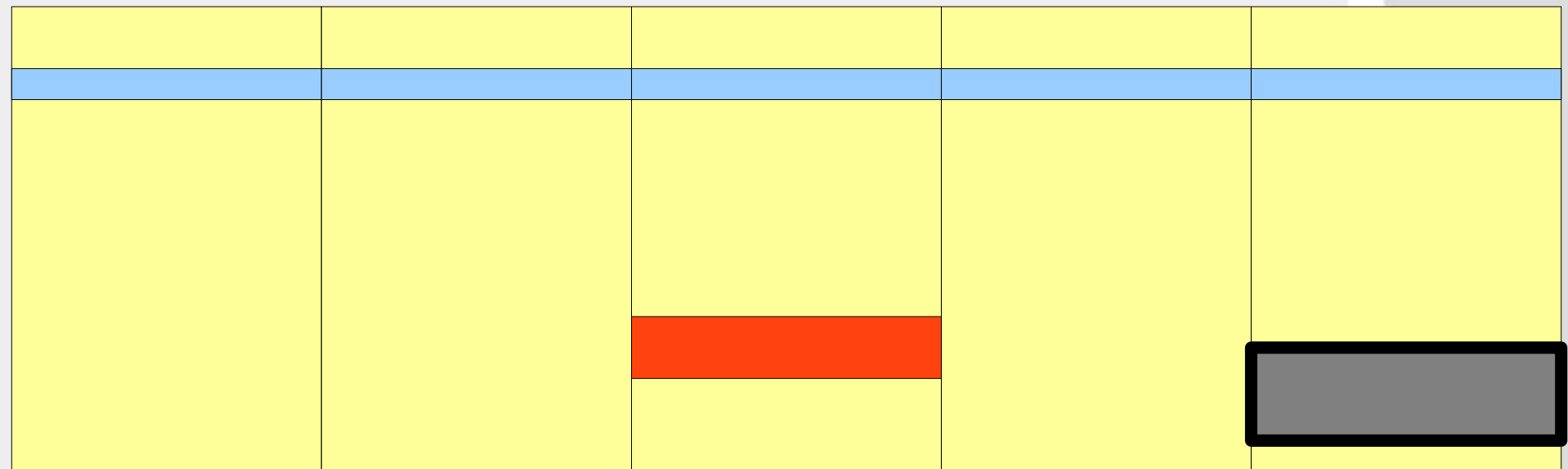
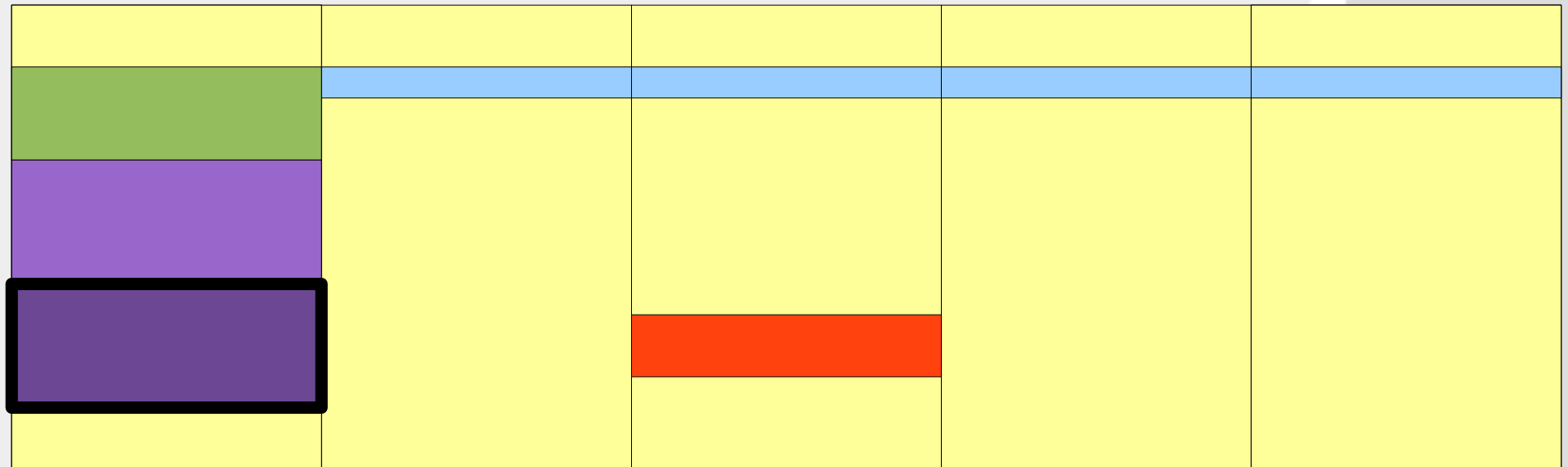
Refactor
→



Emerging Architecture



Zeitplan



- | | | |
|---|---|--|
|  Daily Scrum |  Sprint Planning 1 |  Retrospektive |
|  Review |  Sprint Planning 2 |  Backlog-Pflege ("Schätzmeeting") |

Eigenschaften von strategischem Design

Entscheidungen müssen das ganze Team erreichen

Der Entscheidungsprozess muss Feedback erlauben

Der Plan muss Evolution erlauben

Architekturteams dürfen nicht die besten Entwickler aus den Projekten abziehen

Strategisches Design braucht Minimalismus und Bescheidenheit

Objekte sind Spezialisten, Teammitglieder sind Generalisten



Erlauben uns, unsere Meinung zu ändern

Die perfekte Entscheidung zu treffen ist oft zu teuer oder unmöglich!

Notwendiges „Sicherheitsnetz“ bei inkrementellen Änderungen

Iterative Entwicklung erlaubt es Probleme schnell zu erkennen

Dann: Entscheidung rückgängig machen oder Refactoring

Wichtig ist eine Vorwärtsbewegung, nicht dass alle Entscheidungen richtig sind

Wann werden Entscheidungen getroffen?

Das einfachste mögliche System bauen

Gall's Law

YAGNI – You Aren't Going to Need It

So viele Entscheidungen wie möglich reversibel machen

Entscheidungen so spät wie möglich treffen

Vor allem irreversible Entscheidungen

Änderungen werden oft kommen → klein halten

Continuous Refactoring

Test Driven Development

Domain Driven Design

The best architectures, requirements, and designs emerge from self-organizing teams.

Continuous attention to technical excellence and good design enhances agility.



Danke für Ihre Aufmerksamkeit!



Backup



Frühes und andauerndes
Feedback

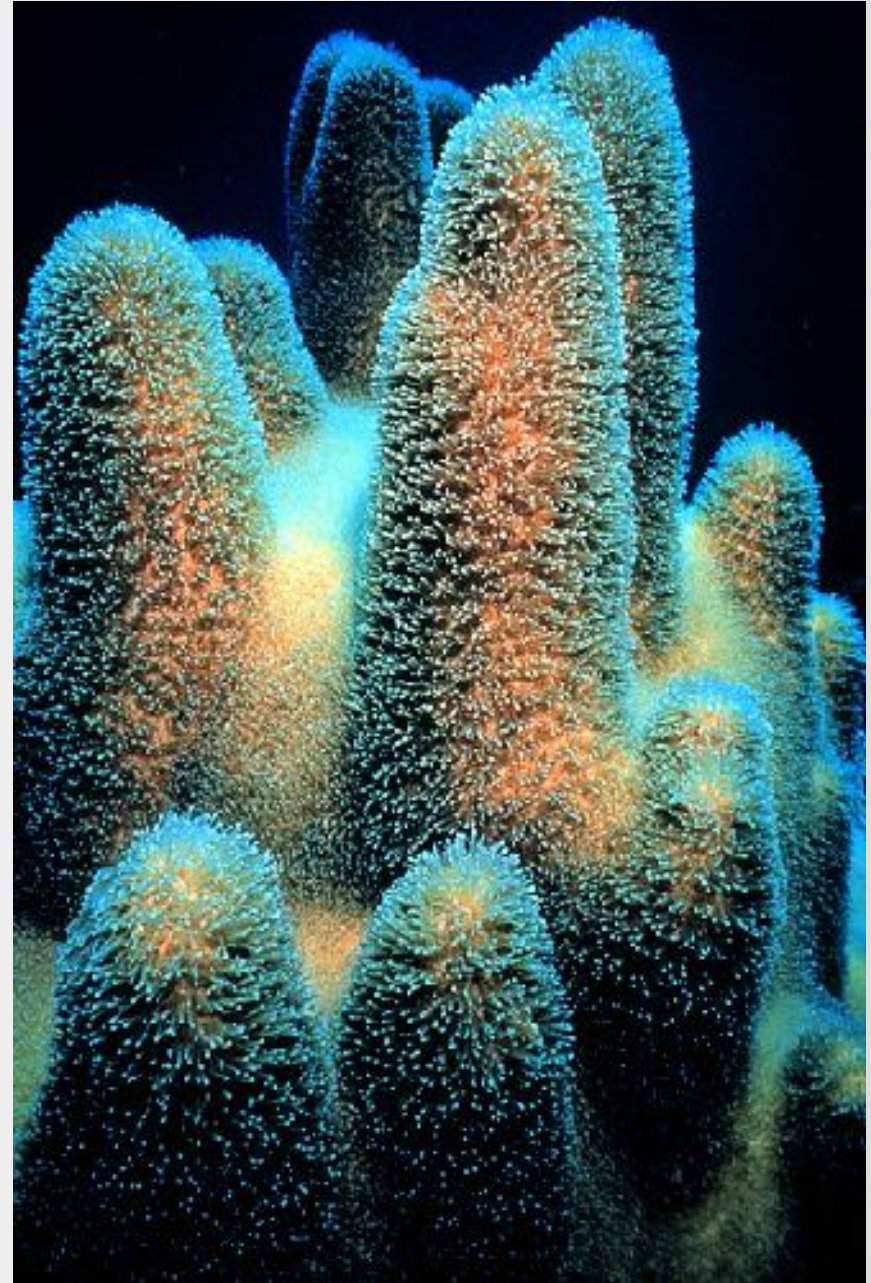
„Fail Fast“

Fehler zulassen

Aber auch korrigieren

Und nicht wiederholen

„Inspect and adapt“



Evolution ist der beste Architekt

Evolutionär: Das Design das überlebt

Der richtige Detaillierungsgrad zur richtigen Zeit

Das einfachste mögliche System

„Spike Solutions“

Kurze Feedback - Zyklen



Idealerweise: Kein Design vorab!

Neuen Scrum – Teams wird das schwer fallen

Dann: Vorab so wenig wie möglich machen

Nach jedem Sprint „Business Value“ liefern

Nach jedem Sprint lauffähige Software liefern

Integriert

In Produktionsqualität