Agil Project Management

Naki Ali, Tima

Betreuer: Matthias Strljic

Table Of Contents

Introduction

Objectives & Research Questions

Scope:

- Apply Agile PM to automated AAS–EDC synchronization (Stay-in-Sync) in IoT contexts

Aims:

- Adapt sprint planning & backlog management for mixed software and configuration workflows
- Introduce Kanban WIP-limits to prevent sync-pipeline bottlenecks
- Compare Scrum, Kanban & hybrid (Scrumban) approaches for our team

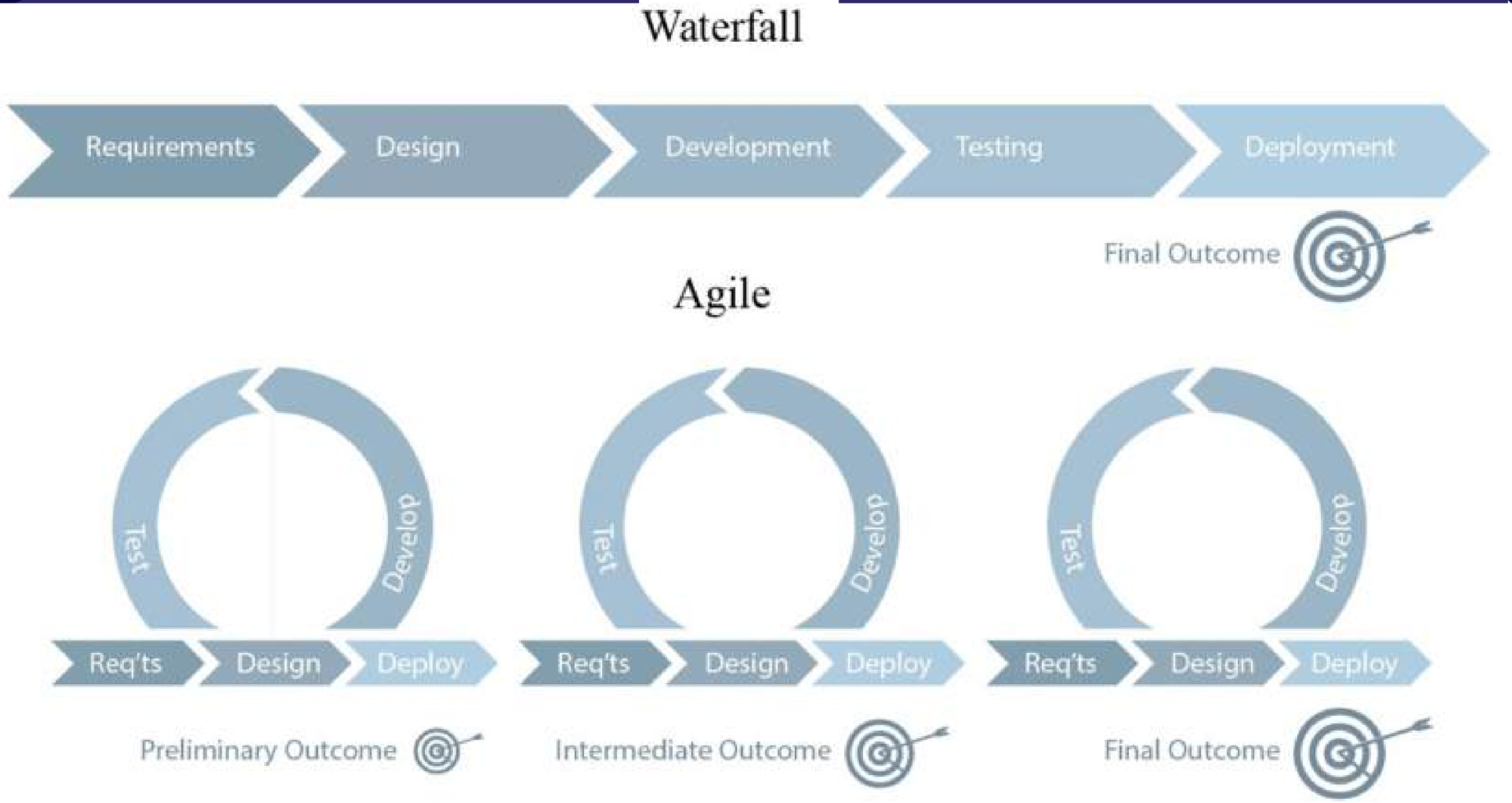
Research Questions:

1. How do 1–2-week sprints affect release frequency & pipeline stability?
2. What impact do explicit WIP-limits have on synchronization error rates?
3. When does Scrumban outperform pure Scrum or Kanban in adaptability and quality?

From Classical to Agile Project Management

Classical PM

Agile PM



Definition

What Is Agile Project Management?

Cooperating with the customer is more important than contract negotiation

People and interactions are more important than processes and tools

Working software is more important than extensive documentation

Responding to changes is more important than following a plan

The Agile Principles

Welcome changing requirements, even late in development.

Deliver working increments frequently, from a couple of weeks to a couple of months.

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

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

Agile methods in project management

Scrum

Iterative Sprints: (1–2 weeks): Fixed timeboxes for planning, development, review & retrospective.

Roles & Ceremonies: Product Owner, Scrum Master, Development Team + Daily Stand-up, Sprint Planning, Sprint Review, Retrospective

Strengths: High predictability, clear responsibilities, regular stakeholder feedback

Kanban

Continuous Flow: No sprint rhythm; tasks are “pulled” as capacity becomes available

WIP Limits: Explicit caps on in-progress work prevent overload and surface bottlenecks immediately

Strength: Low overhead, high flexibility, instant visibility of blockers

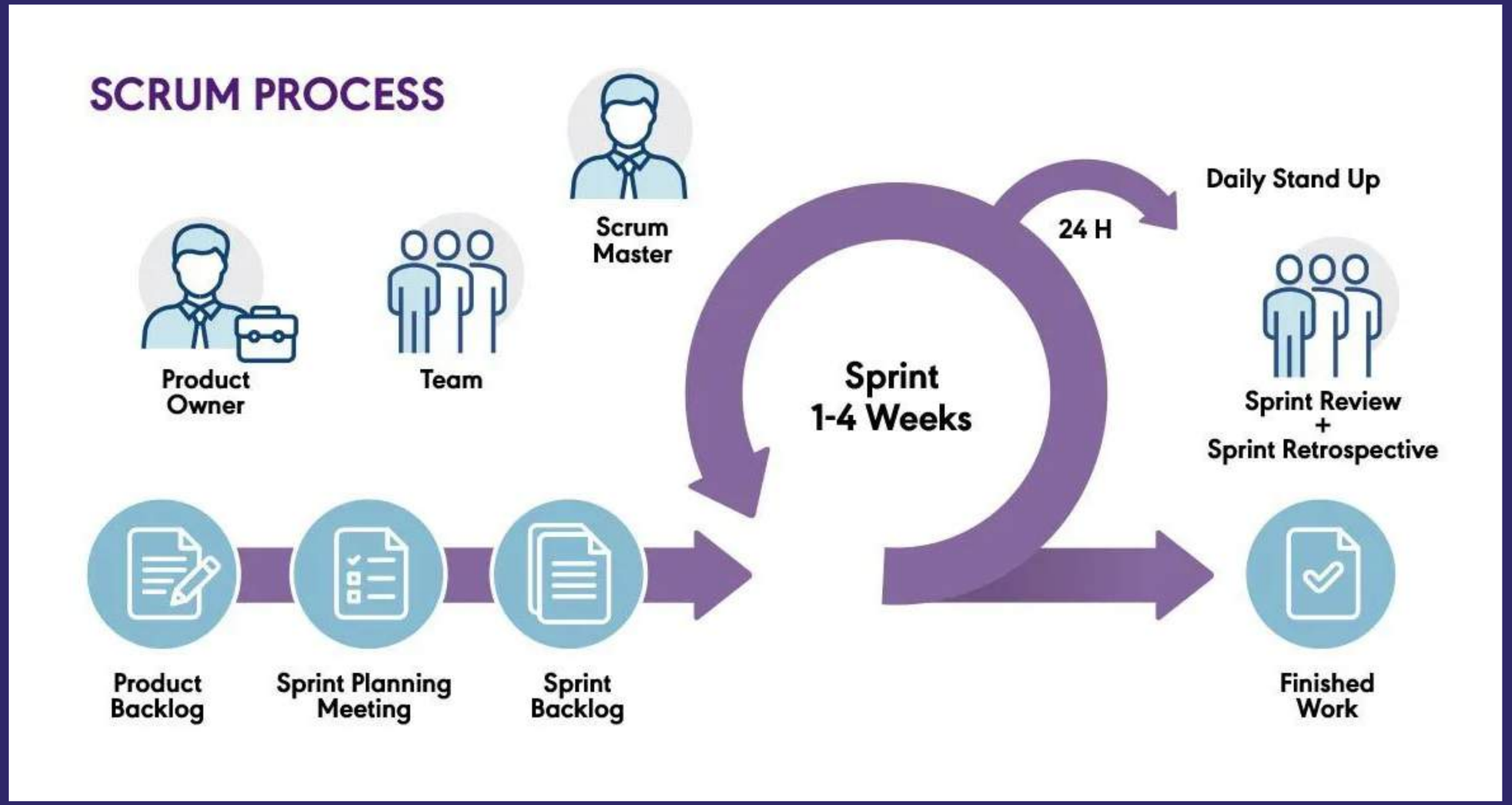
Srumban

Hybrid Approach: 1–2 week sprints for reviews and retrospectives, combined with Kanban-style pull flow during the sprint

Lightweight Planning: Sprint Planning focused on top priorities, on-demand backlog grooming

Strengths: Optimal balance between predictability and flow control, regular feedback cycles with high throughput

Scrum


SCRUM PROCESS
Scrum Master
Product Owner
Team
Daily Stand Up
24 H
Sprint 1-4 Weeks
Sprint Review + Sprint Retrospective
Product Backlog
Sprint Planning Meeting
Sprint Backlog
Finished Work

Scrum

Product Backlog (PO)

All requirements, features, and bugs are collected and prioritized by the Product Owner in a single backlog.

Sprint Planning (PO, SM & Team)

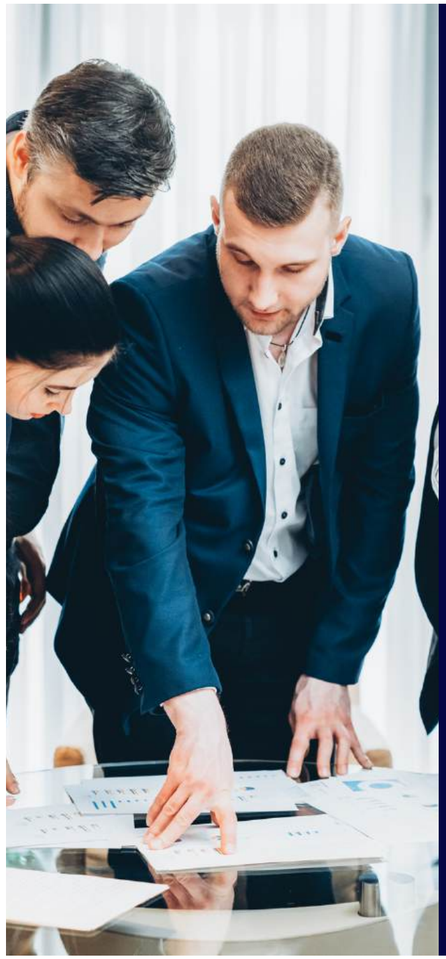
The team defines a sprint goal and selects top-priority backlog items, creating the Sprint Backlog.

Sprint Backlog (Team)

The Development Team breaks each item into actionable tasks within the Sprint Backlog.

Sprint (Team)

Over a fixed timebox (1–4 weeks), the team delivers a potentially shippable increment; the Scrum Master removes blockers.



Daily Scrum (Team & SM)

All requirements, features, and bugs are collected and prioritized by the Product Owner in a single backlog.

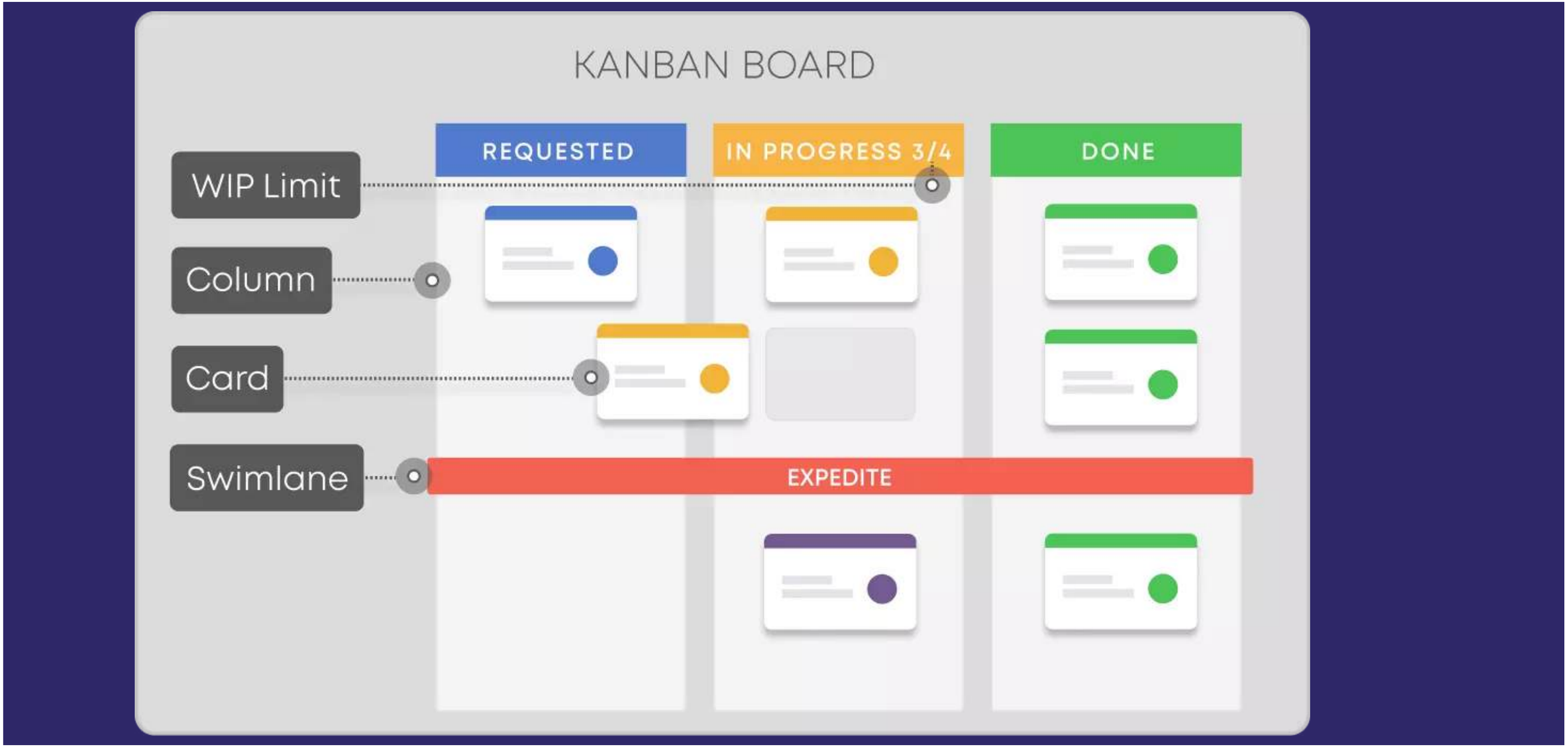
Sprint Review (PO, Team & Stakeholders)

The team demos the increment to the Product Owner and stakeholders; feedback goes back into the backlog.

Sprint Retrospective (PO, SM & Team)

The team reflects on successes and improvements, then defines actions for the next sprint.

Kanban


KANBAN BOARD
REQUESTED
IN PROGRESS 3/4
DONE
WIP Limit
Column
Card
Swimlane
EXPEDITE

Kanban

Backlog & Board Setup

All work items are collected in a backlog and visualized on a Kanban board with columns like To Do, In Progress, Review, and Done.

WIP Limits (Team)

The team defines explicit Work-in-Progress limits for each column to prevent overload and highlight bottlenecks.

Pull-Based Execution (Team)

Developers pull the next highest-priority task into In Progress as soon as they have capacity, ensuring a smooth flow.

Daily Kanban Stand-up (Team & Coach)

A brief daily meeting to review board status, discuss flow metrics, and unblock impediments.

Method Comparison

Comparing Scrum, Kanban und Srumban

CRITERION	CADENCE	PLANNING OVERHEAD	WIP CONTROLS	ROLES & EVENTS	FLEXIBILITY	FEEDBACK LOOPS
SCRUM	2-week Sprintst	High: Sprint Planning & Refinement	Indirect (Sprint scope)	Defined (PO, SM, Dev) + 5 ceremonies	Moderate	Scheduled (Review & Retro each Sprint)
KANBAN	Continuous flow	Low: On-demand prioritization	Direct (explicit WIP limits)	Informal (no mandatory roles/ceremonies)	High	Continuous (ad hoc)
SCRUMBAN	Sprint rhythm + pull-based flow	Medium: Lite Sprint Planning + backlog grooming	Direct (WIP limits within Sprint)	Optional SM + Sprint Review/Retro	Very high	Both scheduled and continuous

Solution 1:

ROLLEN

MEETINGS

LISTEN

PRODUCT OWNER
Matthias Strljic

Der Product Owner Matthias vertritt im Stay-in-Sync-Projekt alle internen und externen Stakeholder und übersetzt deren Anforderungen in konkrete Backlog-Einträge. Im Product Backlog legt er die Prioritäten fest – von AAS-Authentifizierung und Fehlerbehandlung über Sync-Regeln bis hin zu UI-Optimierungen –, sodass das Team immer weiß, welche Teilprodukte als nächstes geliefert werden.

Sammeln von Beiträgen aller Stakeholder

SCRUM MASTER
Max und Junes

Max und Junes unterstützen Matthias und das Entwicklungsteam im Stay-in-Sync-Projekt. Sie sorgen dafür, dass Scrum-Prinzipien und -Zeremonien verstanden und korrekt angewendet werden. Zudem identifizieren und beseitigen sie fortlaufend alle Impediments – etwa fehlende Zugriffsrechte auf Testdaten, blockierende Merge-Requests oder unklare Anforderungen –, damit das Team reibungslos an den geplanten Inkrementen arbeiten kann.

DEVELOPER

Das selbstorganisierende Team – bestehend aus Alex, Metahan, Ruben, Yarden, Ammar, Yasin, Serhat, Max, David und Tima – liefert im Sprint die vereinbarten Inkremente: von der Implementierung der AAS-Authentifizierung und der Fehlerbehandlung über die Konfigurator-UI bis zur Visualisierung der Sync-Flows. Gemeinsam mit Product Owner (Matthias) und Scrum Mastern (Max & Junes) bilden sie das Scrum Team, das alle erforderlichen Fähigkeiten besitzt, um unsere Sync-Pipeline in jedem Sprint voranzubringen.

STAKEHOLDER

Stakeholder sind diejenigen, die Interesse an dem Projekt haben oder es beeinflussen können.

DAILY SCRUM
(max. 15 Min.) Ein tägliches Meeting mit den Developern.. Das Meeting kann strukturiert werden basierend auf den folgenden drei Fragen:
1. Was habe ich gestern gemacht?
2. Was werde ich heute tun?
3. Gibt es irgendwelche Hürden?

SPRINT REVIEW
(max. 4 Stunden bei vierwöchigen Sprints, bei kürzeren Sprints entsprechend kürzer) Am Ende des Sprints präsentiert das Entwicklungsteam die Ergebnisse und bekommt von den Beteiligten Feedback. Das können beispielsweise Kunden, Vorgesetzte und/oder Kollegen sein.

PRODUCT BACKLOG REFINEMENT
(max. 10% der verfügbaren Arbeitszeit für die Pflege und Aufbereitung des Backlogs) Im Refinement Meeting stellt das Entwicklerteam Fragen und schätzt den Umfang.

SPRINT

Los geht's!

Ein Sprint dauert 1 bis 4 Wochen

SPRINT PLANNING-1,2,3
(max. 5% der verfügbaren Arbeitszeit)
Abschnitt 1: Der Scrum Master lässt den Product Owner das Ziel des Sprints sowie den Product Backlog erläutern. Er macht deutlich, dass weitere Sprints folgen werden und bittet die Teilnehmer um ihre Zusage. Das Team wählt User Stories aus dem Product Backlog aus, die im nächsten Sprint angegangen werden sollen.
Abschnitt 2: Der Product Owner bespricht das Sprint-Ziel und die Stories, mit denen dieses Ziel erreicht werden soll. Die Developer wählen dann den Umfang der Arbeit für den bevorstehenden Sprint.
Abschnitt 3: Dieser Teil bestimmt wie die Arbeiten operativ durchgeführt werden sollen.

SPRINT RETROSPECTIVE
(max. 2% der verfügbaren Arbeitszeit)
Der Schwerpunkt der Retrospektive liegt auf Team- und und Prozessverbesserung.
Zentrale Fragen sind:
1. Was lief gut während des Sprints?
2. Was kann verbessert werden für den nächsten Sprint?

USER STORY
Bei einer User Story handelt es sich um eine kurze Beschreibung eines Teils eines Produktes. Sie ist aus Sicht des Endnutzers verfasst. Eine User Story hat den folgenden Aufbau: Als (wer?) möchte ich (was?), damit ich (warum?).

PRODUCT BACKLOG
Eine Liste mit allen Arbeiten, die zu verrichten sind, um ein Produkt zu entwickeln und zu pflegen. Der Product Owner verwaltet den Product Backlog und versieht ihn mit Prioritäten. Die Issues welche im Stay-in Sync GitHub hinterlegt sind.

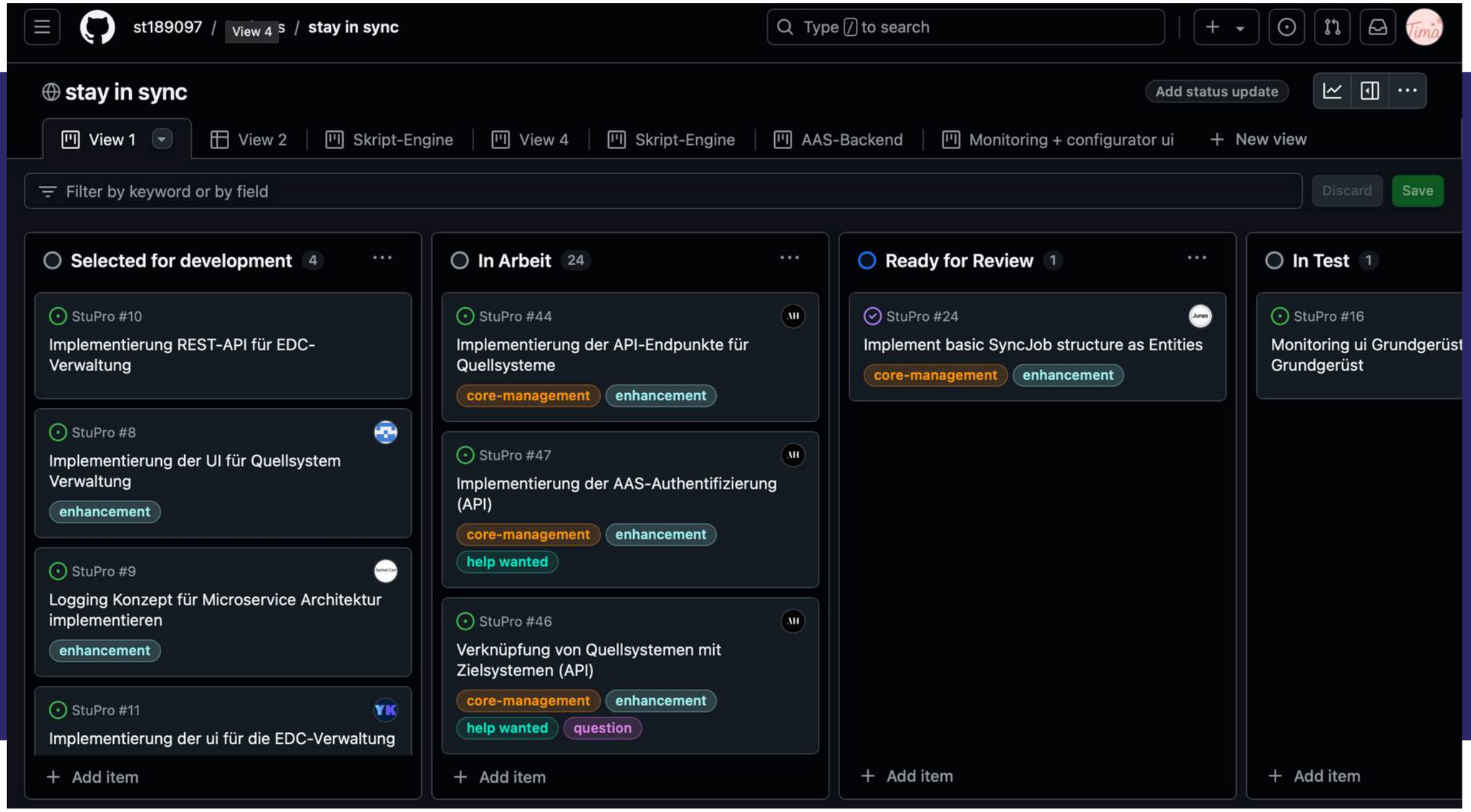
SPRINT BACKLOG
Eine Übersicht der Arbeiten aus dem Product Backlog, die das Entwicklungsteam für den bevorstehenden Sprint ausgewählt hat. Der Arbeitsplan folgt einem Arbeitsablauf: To Do / Doing / Done

ARBEITSPAKETE BESTIMMEN
Die Aufteilung des ausgewählten Product Backlog Items in überschaubare Arbeitsabschnitte. Die Aufgaben können im Prinzip innerhalb eines Arbeitstages oder weniger erledigt werden.

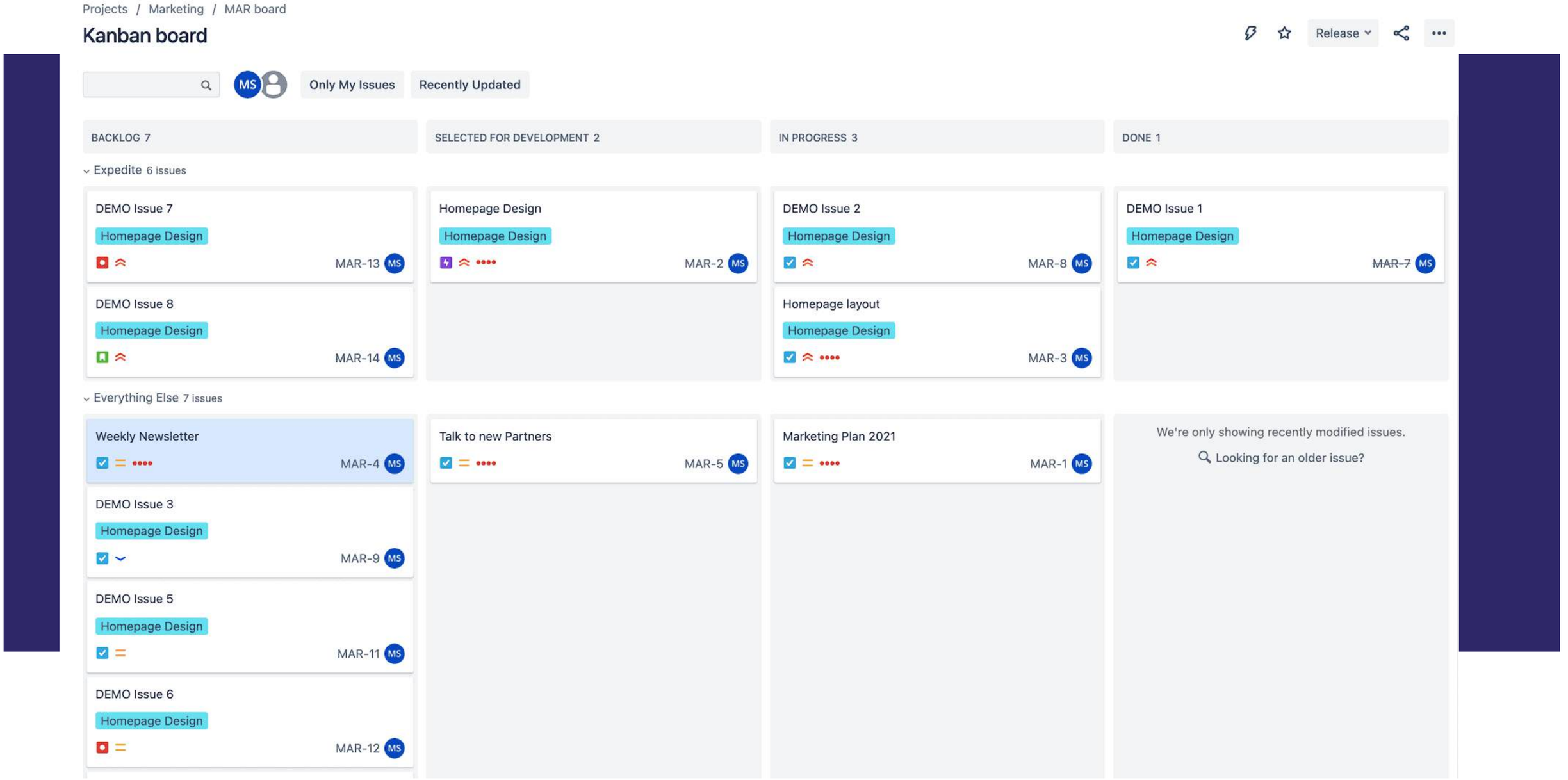
BURN DOWN CHART
Das Sprint Burn Down Chart zeigt den verbleibenden Arbeitsaufwand des aktuellen Sprints. Das Chart wird von den Entwicklern gepflegt.

DEFINITION OF DONE
Eine Liste mit Anforderungen, die ein Item aus dem Sprint Backlog zu erfüllen hat, damit es als erledigt gilt. Sie wird regelmäßig von dem Scrum Team überprüft.

Solution 2:


st189097 / View 4 s / stay in sync
Type / to search
stay in sync
Add status update
View 1
View 2
Skript-Engine
View 4
Skript-Engine
AAS-Backend
Monitoring + configurator ui
New view
Filter by keyword or by field
Discard
Save
Selected for development 4
StuPro #10
Implementierung REST-API für EDC-Verwaltung
StuPro #8
Implementierung der UI für Quellsystem Verwaltung
enhancement
StuPro #9
Logging Konzept für Microservice Architektur implementieren
enhancement
StuPro #11
Implementierung der ui für die EDC-Verwaltung
Add item
In Arbeit 24
StuPro #44
Implementierung der API-Endpunkte für Quellsysteme
core-management
enhancement
StuPro #47
Implementierung der AAS-Authentifizierung (API)
core-management
enhancement
help wanted
StuPro #46
Verknüpfung von Quellsystemen mit Zielsystemen (API)
core-management
enhancement
help wanted
question
Add item
Ready for Review 1
StuPro #24
Implement basic SyncJob structure as Entities
core-management
enhancement
Add item
In Test 1
StuPro #16
Monitoring ui Grundgerüst
Grundgerüst
Add item

Solution 3: Jira



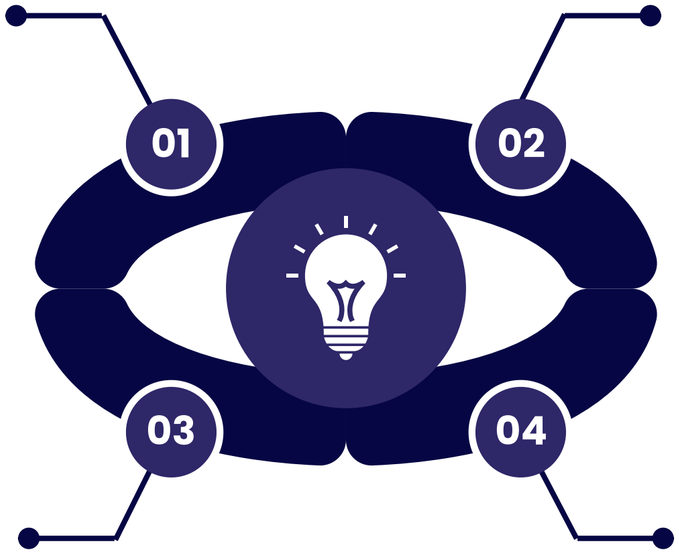
Evaluation Criteria

Planability

- **Scrum**: **High** – fixed 2-week sprints and burndown charts enable reliable scheduling.
- **Kanban**: **Medium** – continuous flow without time boxes; releases need to be scheduled separately.
- **Scrumban**: **High** – Combines sprint timeboxes with pull mechanisms, delivering both predictability and flexibility.

WIP-Control (Work-in-Progress)

- **Scrum**: **Medium** – the sprint-scope limitation controls how many tasks can be worked on in a sprint, but lacks explicit column limits.
- **Kanban**: **High** – explicit WIP limits in each column prevent overload and make bottlenecks immediately visible.
- **Scrumban**: **High** – leverages both WIP limits (Kanban) and sprint scope (Scrum) for maximum transparency.



Flexibility

- **Scrum**: **Medium** – changes during a sprint are limited; new feedback only flows into the next sprint.
- **Kanban**: **High** – the team pulls new issues immediately as capacity frees up; urgent hotfixes hit the board without delay.
- **Scrumban**: **High** – allows a pull flow for spontaneous tasks while retaining a sprint rhythm for greater predictability.

Team Maturity & Role Understanding

- **Scrum**: **High** – requires the PO, SM, and team to understand their roles and routinely execute Scrum ceremonies.
- **Kanban**: **Medium** – lower process overhead, suitable for beginners, but demands discipline in enforcing WIP limits.
- **Scrumban**: **Medium** – requires moderate process knowledge and is quickly adoptable by experienced teams.

Conclusion

Recommendation:

- Hybrid (Scrumban) is the best fit for Stay-in-Sync:
  - Combines predictability of 2-week sprints with flexibility of pull-based flow
  - Enforces WIP limits and regular feedback via reviews & retrospectives
- Tooling:
  - Jira as unified platform: Scrum boards, Kanban boards, automations, burndown & flow charts

Next Steps:

1. Pilot Scrumban Sprint (2 weeks)
   - Set up hybrid board in Jira: columns “To Do / In Progress (WIP≤3) / Review / Done” + Sprint Backlog
   - Create templates for user stories, tasks & Definition of Done
2. Team Onboarding
   - 1-hour workshop: roles, board workflow, ceremonies
   - Assign Jira permissions & automation rules
3. Establish Metrics & Cadence
   - Track burndown and cumulative flow
   - Weekly Delivery Review meeting
   - Bi-weekly Retrospective
4. Inspect & Adapt
   - After pilot, gather feedback, refine WIP limits, adjust board & ceremonies
   - Roll out adjusted Scrumban process to all teams

References

- https://www.microtool.de/en/knowledge-base/what-is-agile-project-management/
- https://www.spf-consulting.com/en/insights/how-to-apply-agile-project-management/
- https://www.awork.com/blog/agile-project-management
- https://ifm-business.de/aktuelles/business-news/hybrides-projektmanagement-definition-und-methoden-von-traditionell-ueber-agil-bis-hybrid.html
- https://www.pm-partners.com.au/insights/the-agile-journey-a-scrum-overview/
- https://agilescrumgroup.de/scrum/
- https://businessmap.io/de/kanban-ressourcen/kanban-erste-schritte/was-ist-kanban
- https://www.adito.de/knowhow/blog/projektmanagement
- https://www.atlassian.com/software/jira
- https://scrumguides.org
- https://medium.com/@sudarhtc/agile-project-management-methodology-manifesto-frameworks-and-process-f4c332ddb779
- https://github.tik.uni-stuttgart.de/users/st189097/projects/2/views/1