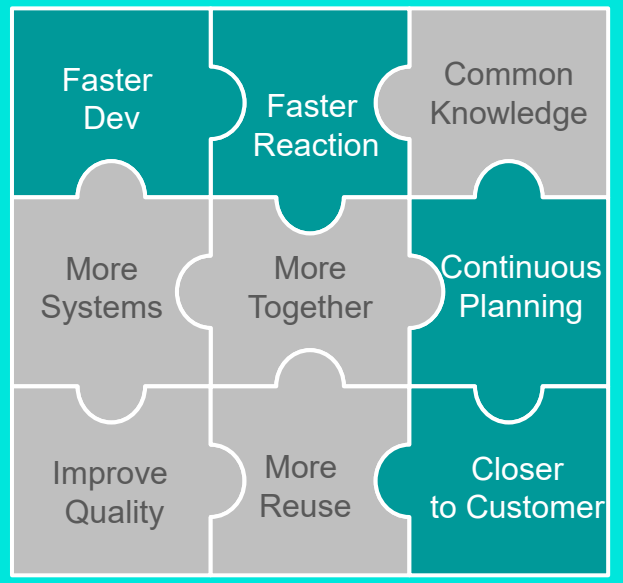
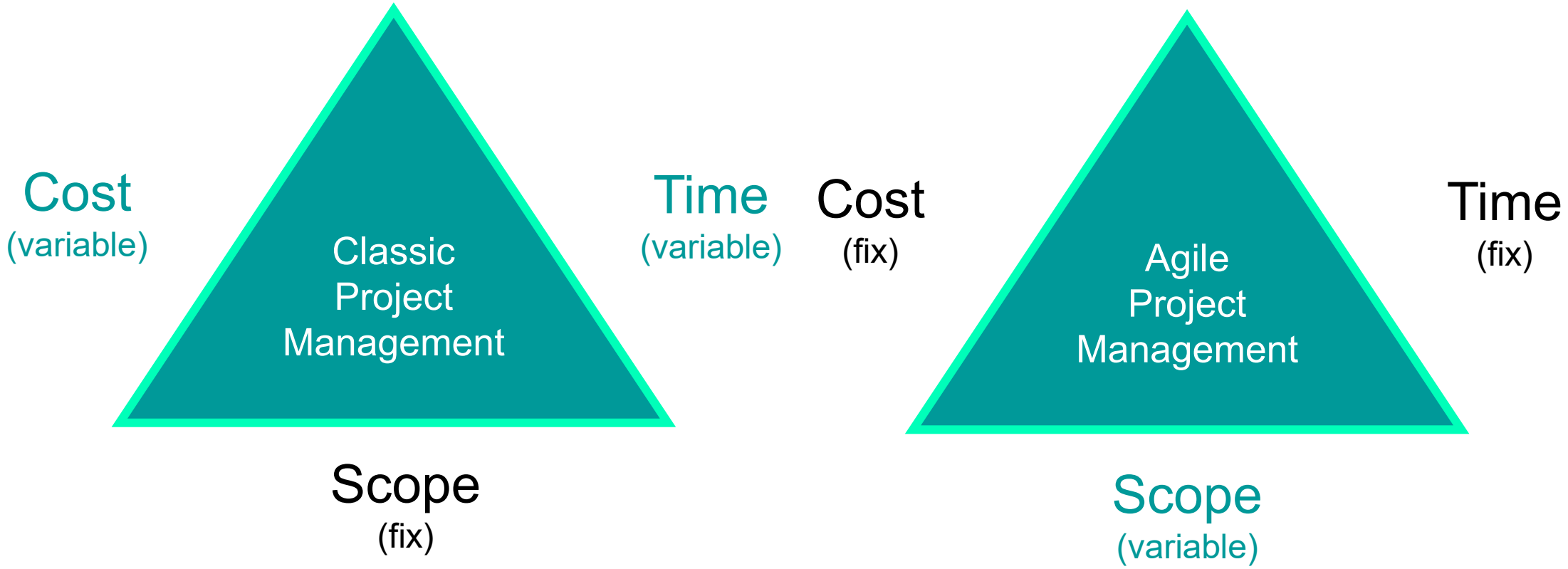


RE in agile Environment

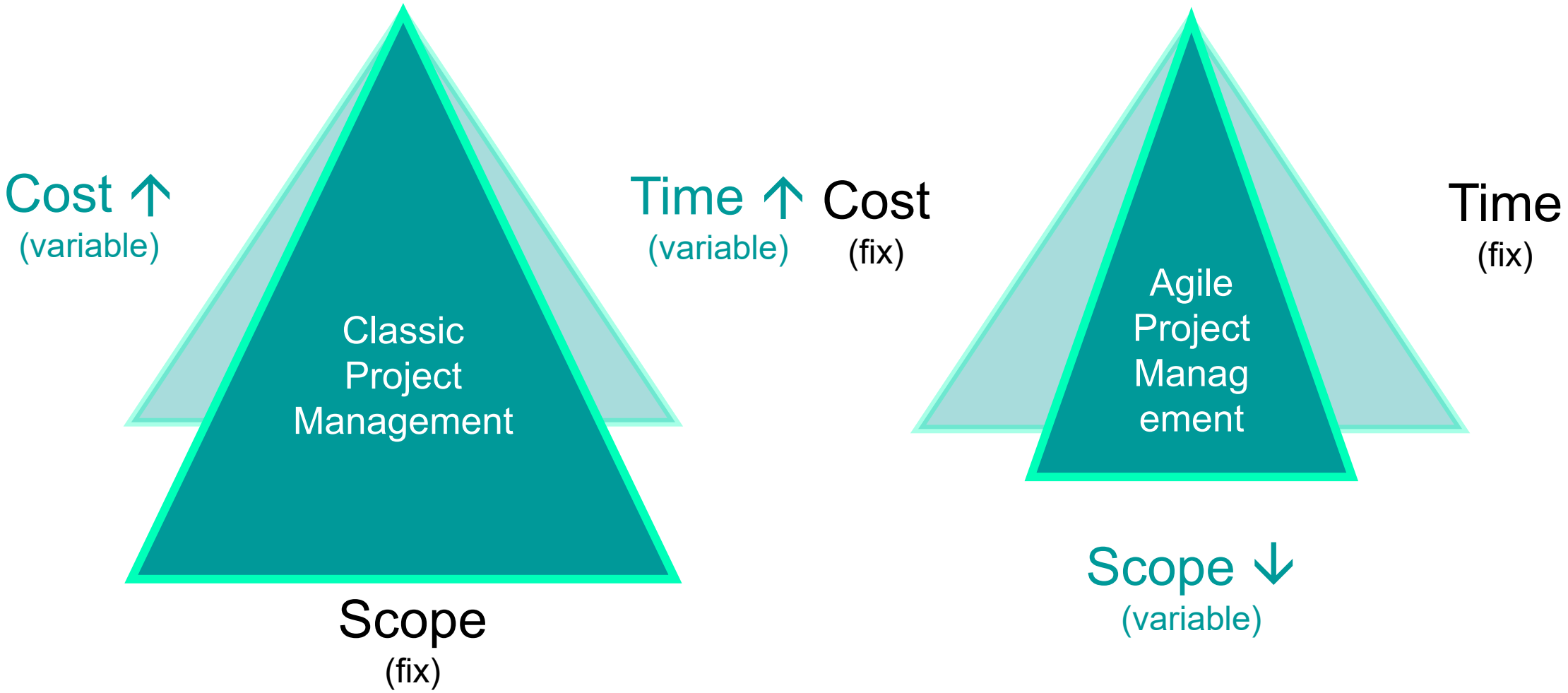
What's the difference to classic RE?

Agility Aspects related to Classical Triple Constraint



SIEMENS

Agility Aspects related to Classical Triple Constraint



SIEMENS

Defining Agile Requirements Engineering (ARE)

Agile Requirements Engineering is an approach that aligns with the Agile methodology, **focusing on iterative development, collaboration, and flexibility**. Unlike traditional requirements engineering, which typically involves extensive documentation and upfront planning, Agile Requirements Engineering **emphasizes adaptability and continuous feedback**.

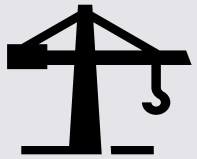
> Stop starting, start finishing

SIEMENS

Comparing Agile Requirements Engineering to Traditional Methods

Traditional Requirements Engineering

- Upfront Planning
- Rigid Process
- Delayed Feedback
- Longer Time-to-Market



Agile Requirements Engineering

- Continuous Planning
- Flexible Process
- Frequent Feedback
- Shorter Time-to-Market

SIEMENS

Comparing Agile Requirements Engineering to Traditional Methods

Traditional Requirements Engineering:

- **Upfront Planning**: Extensive documentation and planning are done at the beginning of the project.
- **Rigid Process**: Changes are difficult to accommodate once the project is in the later stages.
- **Delayed Feedback**: Feedback is typically received at the end of the project, increasing the risk of misalignment with customer needs.
- **Longer Time-to-Market**: The sequential nature of the process often results in longer development cycles.

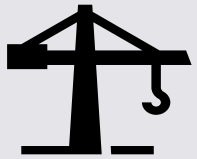
Agile Requirements Engineering:

- **Continuous Planning:** Planning is done iteratively, allowing for ongoing refinement of requirements.
- **Flexible Process:** Changes can be incorporated at any stage of the project.
- **Frequent Feedback:** Regular interactions with stakeholders ensure continuous feedback and alignment.
- **Shorter Time-to-Market:** Incremental delivery enables faster deployment of functional system.

Challenges and overcome challenges in Agile Requirements Engineering

Challenges

- Maintaining Documentation
- Stakeholder Availability
- Scope Creep
- Team Alignment



Techniques to Overcome Challenges

- Living Documentation
- Regular Stakeholder Meetings
- Clear Definition of Done
- Strong Communication Channels

SIEMENS

Challenges and overcome challenges in Agile Requirements Engineering

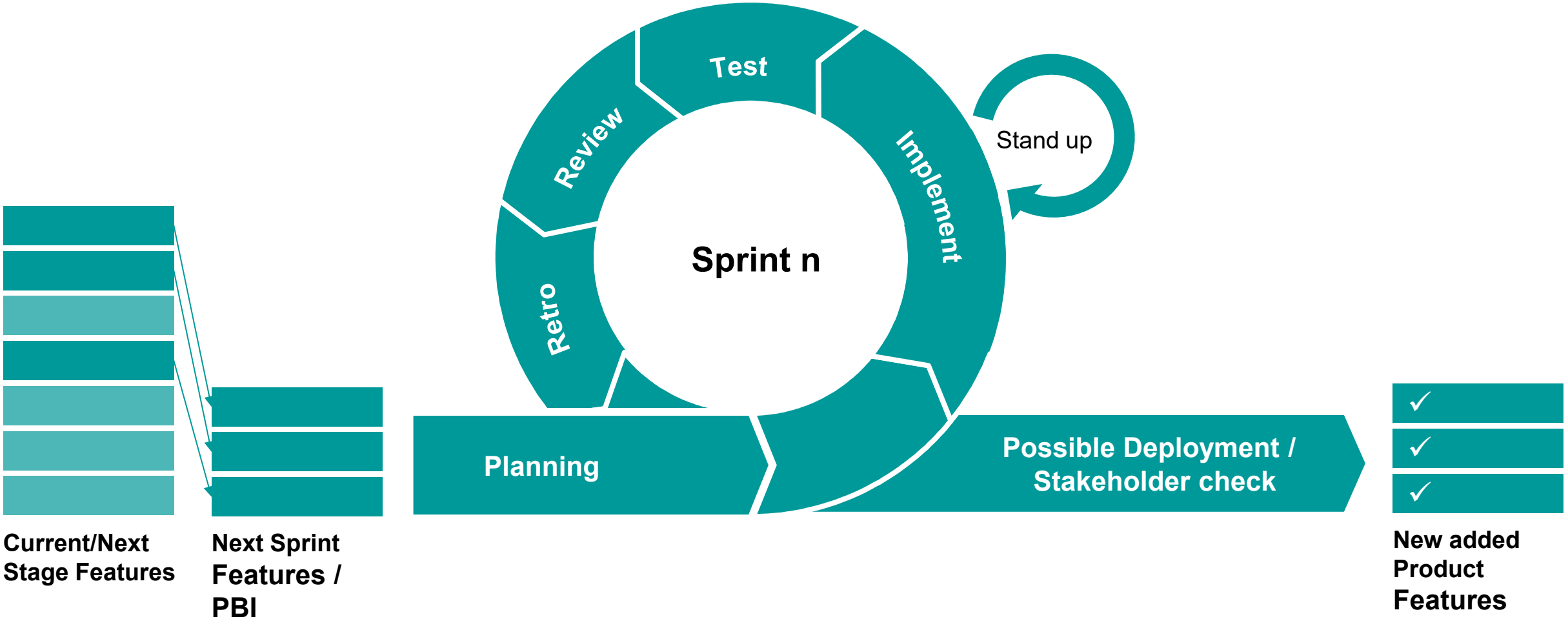
Challenges

1. **Maintaining Documentation**: In Agile, the focus on working systems can sometimes lead to inadequate documentation. Balancing sufficient documentation with the Agile principle of simplicity is crucial.
2. **Stakeholder Availability**: Continuous collaboration requires stakeholders to be actively involved throughout the project. Ensuring their availability can sometimes be challenging.
3. **Scope Creep**: The flexibility to accommodate changes can sometimes lead to scope creep, where additional features are continuously added, potentially impacting timelines and budgets.
4. **Team Alignment**: Ensuring that all team members have a shared understanding of the requirements and priorities requires effective communication and coordination.

Techniques to Overcome Challenges

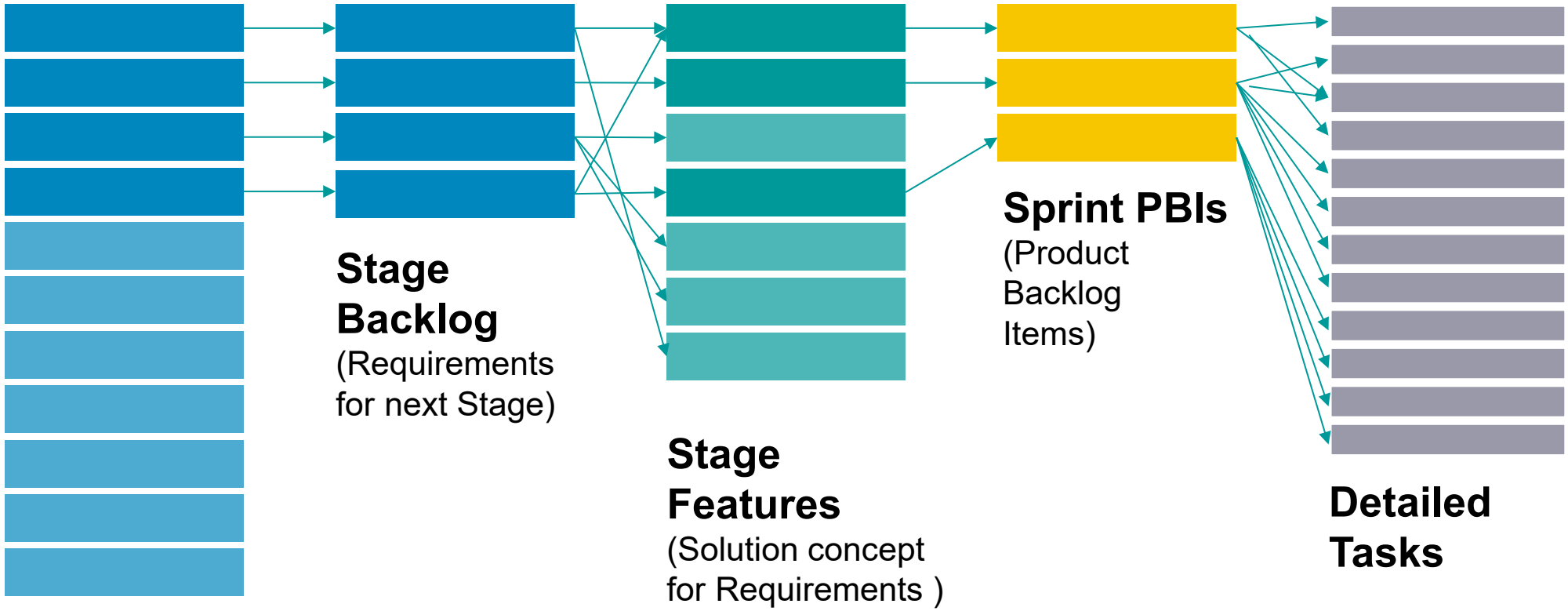
1. **Living Documentation**: To maintain adequate documentation without hindering agility, teams can adopt the concept of "living documentation." This means keeping documents up-to-date with the latest developments and using tools that automatically generate documentation from code and tests.
2. **Regular Stakeholder Meetings**: Scheduling regular meetings with stakeholders, such as weekly or bi-weekly check-ins, can ensure their continuous involvement and timely feedback.
3. **Clear Definition of Done**: Establishing a clear and agreed-upon definition of done for user stories can help mitigate scope creep. This ensures that all team members and stakeholders have a shared understanding of when a requirement is considered complete.
4. **Strong Communication Channels**: Utilizing tools and practices that enhance communication, such as collaborative platforms, regular stand-ups, and pair programming, can help align team members and ensure everyone is on the same page.

Sprint approach (according to scrum methodology)



SIEMENS

How get from Product Requirements to Sprint Tasks



Product Backlog
(Requirements for Release 0.8)

Stage Backlog
(Requirements for next Stage)

Stage Features
(Solution concept for Requirements)

Sprint PBIs
(Product Backlog Items)

Detailed Tasks

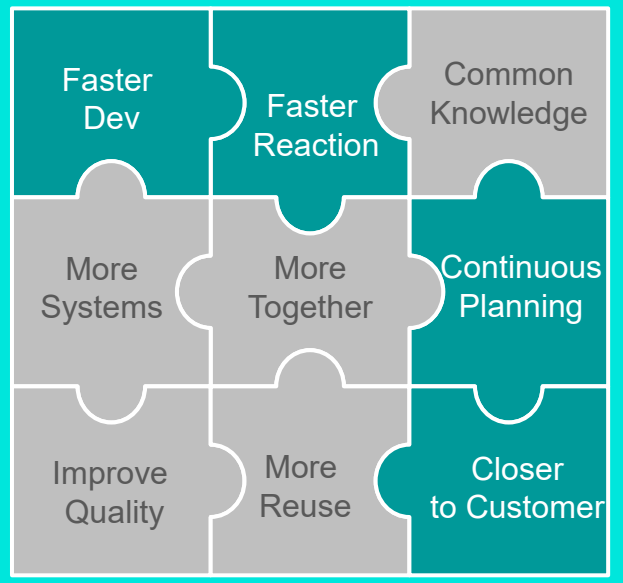
All lists are ranked, so everybody knows, what's next.

SIEMENS

Benefits of Agile Requirements Engineering (ARE)

1	Enhanced Flexibility	ARE allows teams to adapt to changes quickly, ensuring that the final product aligns with the evolving needs of the customer.
2	Improved Stakeholder Engagement	Continuous collaboration with stakeholders ensures that their needs are met, leading to higher satisfaction and better alignment with business goals.
3	Early Detection of Issues	By delivering incremental updates, teams can identify and address issues early in the development process, reducing the risk of major setbacks.
4	Faster Time-to-Market	The iterative nature of ARE enables the delivery of functional increments sooner, providing value to the customer more quickly.
5	Better Quality	Continuous feedback and iterative improvements lead to higher quality system that meets the actual needs of users.

SIEMENS



RE in agile Environment

What's the difference to classic RE?

Q&A

SIEMENS