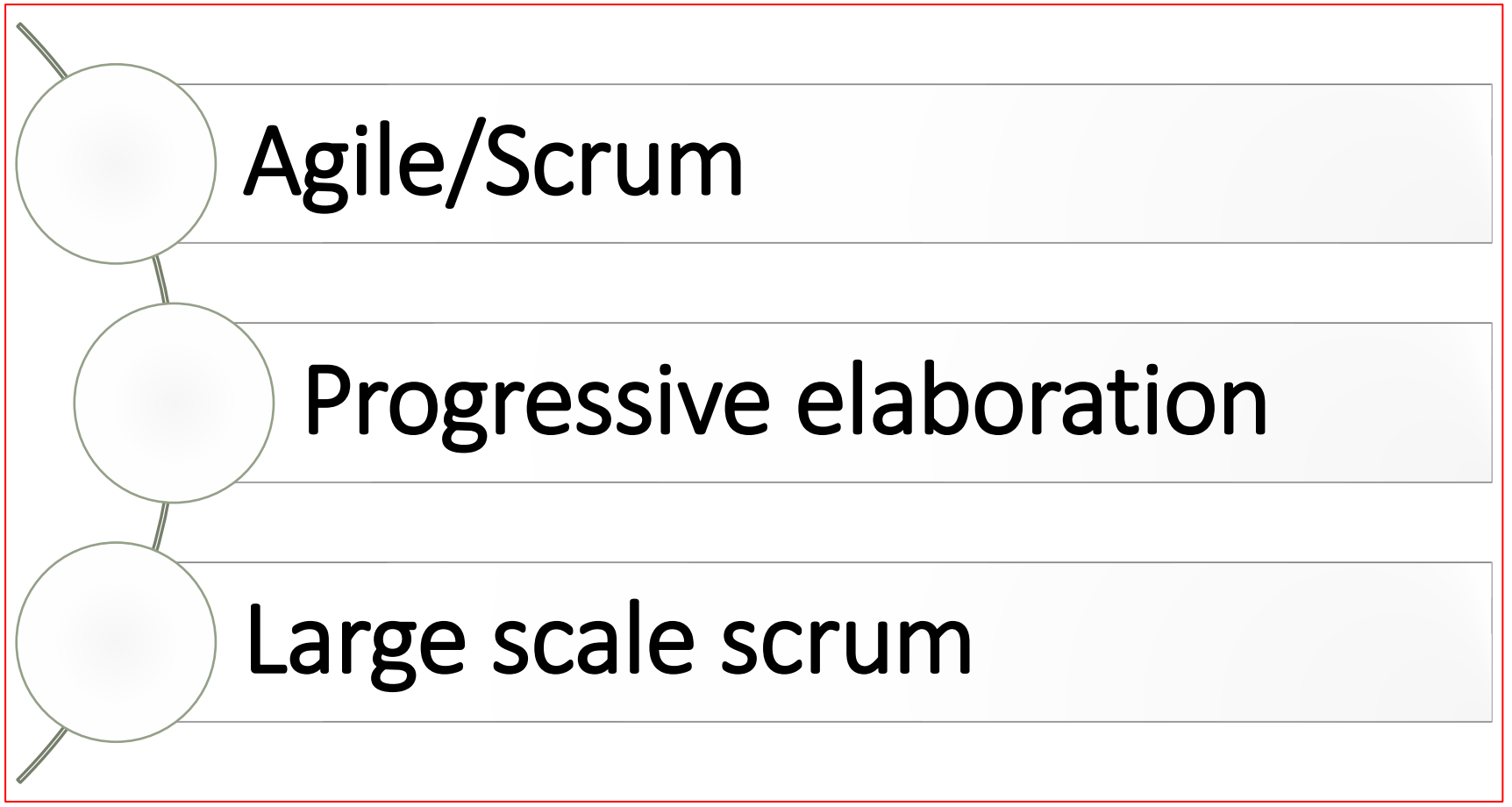
iPMAC
NET YOUR WORK

DEVSECOPS COURSE

SOFTWARE DEVELOPMENT FRAMEWORK

TRAINER: TRAN HUU HOA

AGENDA



AGILE / SCRUM

Agile

An approach to project management that emphasizes incremental, iterative steps for completing projects

Iterative and Incremental: Agile breaks down projects into smaller, manageable pieces called sprints. After each sprint, teams reflect and adjust their strategy for the next one

AGILE / SCRUM

Agile - 12 principals:

1. *Customer Satisfaction*: Deliver valuable software that satisfies customers.
2. *Welcome Change:* Adapt to changing requirements throughout the project.
3. *Frequent Delivery:* Deliver working software frequently (in short iterations).
4. *Collaboration*: Foster collaboration between developers, stakeholders, and users.
5. *Motivated Individuals*: Trust and empower motivated team members.
6. *Face-to-Face Communication*: Prioritize direct communication over written documentation.
7. *Working Software:* Measure progress by functional software, not just plans.
8. *Sustainable Pace*: Maintain a sustainable work pace to avoid burnout.
9. *Technical Excellence:* Focus on quality, good design, and technical practices.
10. *Simplicity*: Strive for simplicity in design and processes.
11. *Self-Organizing Teams*: Trust teams to make decisions and adapt.
12. *Reflection and Adaptation*: Regularly reflect on processes and adjust as needed.

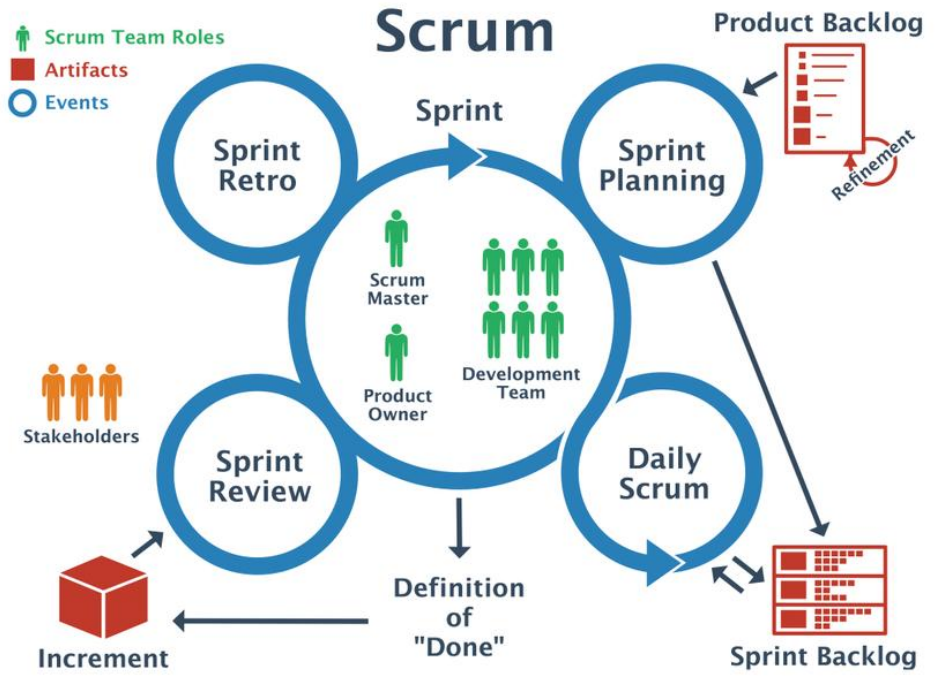
AGILE / SCRUM

Scrum:

An agile framework commonly used in software development and other industries. Scrum is a lightweight yet powerful framework that helps teams self-organize and deliver products and services in short cycles

Structure:

- **Scrum Team**: Consists of a Product Owner, a Scrum Master, and Developers. Each has specific accountabilities.
- **Events**: Scrum Teams participate in five events, including Sprints (short cycles of one month or less).
- **Artifacts**: Product Backlog, Sprint Backlog, and the Increment.



AGILE / SCRUM

Scrum

- *Inspiration:* The term “Scrum” is inspired by the rugby formation where the team comes together to move the ball forward. Similarly, in Scrum, the team collaborates to move the product forward.
- *Empiricism:* Scrum is empirical, based on observation, experience, and experimentation. It emphasizes working iteratively and adapting as needed.
- *Pillars of Scrum:*
  - Transparency: All work and processes are visible.
  - Inspection: Regularly review progress and adapt.
  - Adaptation: Make changes based on feedback.
- *Scrum Values:* Courage, Focus, Commitment, Respect, and Openness guide Scrum Teams in their collaboration

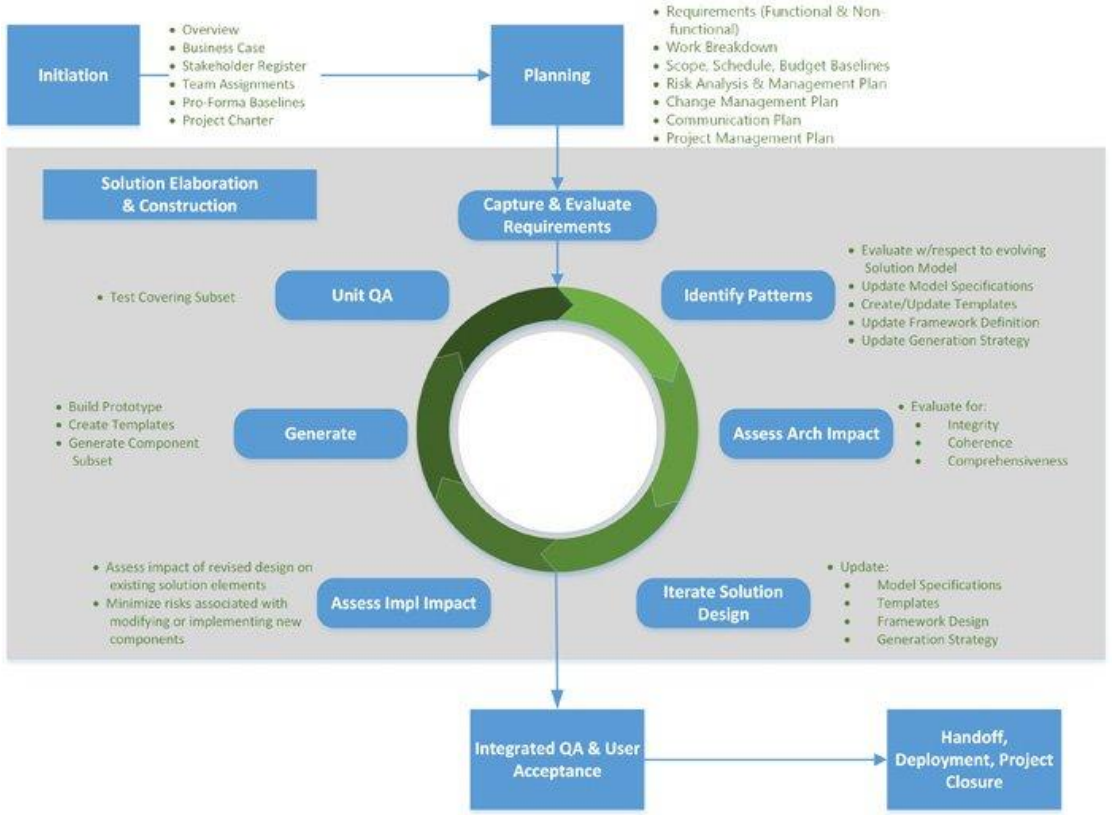
PROGRESSIVE ELABORATION

Progressive elaboration in project management refers to the continuous refinement and improvement of a project plan as new insights and information emerge throughout the project's lifecycle.

Progressive elaboration significantly impacts project success by enhancing adaptability, accuracy, and stakeholder satisfaction:

- *Adaptability:* As projects evolve, new information emerges. Progressive elaboration allows project managers to adjust plans, resources, and timelines accordingly. This adaptability ensures that the project remains aligned with changing requirements, reducing the risk of failure.
- *Accuracy:* Initially, project plans are based on assumptions and limited data. By continuously refining the plan, project teams gain a clearer understanding of scope, risks, and dependencies. This leads to more accurate estimates, resource allocation, and cost projections.
- *Stakeholder Satisfaction:* Regular updates and refinements foster transparency. Stakeholders appreciate being informed about progress, challenges, and adjustments. Their confidence in the project increases, contributing to overall success.

PROGRESSIVE ELABORATION



PROGRESSIVE ELABORATION

- *Cross-Functional Teams*: Assemble diverse teams with varied expertise (developers, designers, testers, etc.). Their different perspectives enhance problem-solving and creativity.
- *Open Communication Channels*: Foster an environment where team members freely share ideas, concerns, and updates. Regular stand-up meetings, chat platforms, and collaborative tools facilitate this.
- *Shared Vision*: Ensure everyone understands the project's purpose, goals, and desired outcomes. A shared vision motivates collaboration and aligns efforts.
- *Iterative Reviews:* Regularly review and refine project artifacts (requirements, designs, etc.). Involve stakeholders to gather feedback and make informed adjustments.
- *Collaborative Tools:* Use tools like version control systems, wikis, and virtual whiteboards. They promote real-time collaboration and knowledge sharing.

LARGE SCALE SCRUM

Large Scale Scrum (LeSS) is a framework for scaling Scrum across multiple teams working on a single product. The goal of LeSS is to apply the principles and methods defined by Scrum to large enterprises as simply as possible. It uses the same roles (product owners, managers, scrum masters, and scrum teams) as single-team Scrum but adapts them to a large-scale context where multiple teams collaborate to produce one shippable product.

LARGE SCALE SCRUM

- *Assess Readiness:* Evaluate your organization's current agile maturity, culture, and willingness to embrace change. Ensure leadership support and commitment.
- *Educate Teams:* Train teams on Scrum principles, roles, and practices. Explain the LeSS framework and its benefits.
- *Form Cross-Functional Teams:* Identify product areas or features. Create cross-functional teams with product owners, developers, and testers.
- *Shared Backlog:* Maintain a single product backlog for all teams. Prioritize items collaboratively.
- *Scrum of Scrums (SoS):* Establish regular SoS meetings. Scrum Masters from each team discuss dependencies, impediments, and alignment.
- *Integrated Increments:* Teams deliver potentially shippable increments. Continuous integration and testing are crucial.
- *Cross-Team Retrospectives:* Conduct retrospectives involving representatives from different teams. Identify improvement areas and foster collaboration.
- *Leadership Support:* Encourage servant leadership. Remove organizational impediments.
- *Adapt and Learn:* LeSS is empirical; inspect and adapt. Continuously improve processes.
- *Patience and Persistence:* Scaling agility takes time. Be patient and persistent.