



Everyone's Smart
University

PMG 622

Agile Project Management

Final Exam Summary

دكتور مهندس استشاري ادارة المشروعات

Eng. Dr. Akram Mustafa Suliman Ali,

PhD. AU, BEng. & MSc. AAST, CEng CMarEng MIMarEST, PE, PMP®,

ASQ CSSBB, ISO Lead Auditor, NDT Level II,



Everyone's Smart
University

PMG 622

Agile Project Management

LECTURE 01: An Introduction to Agile

دكتور مهندس استشاري ادارة المشروعات

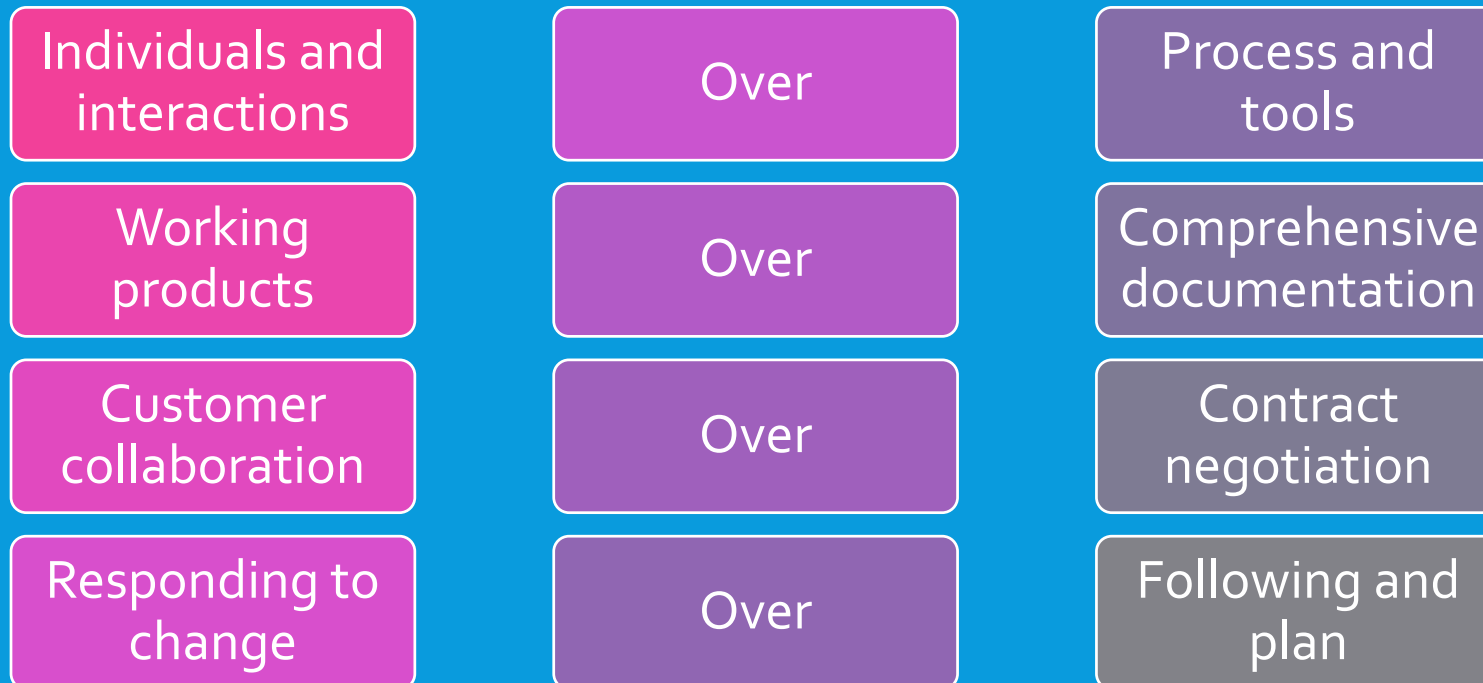
Eng. Dr. Akram Mustafa Suliman Ali,

PhD. AU, BEng. & MSc. AAST, CEng CMarEng MIMarEST, PE, PMP®,

ASQ CSSBB, ISO Lead Auditor, NDT Level II,

AGILE MANIFESTO / “4 VALUES”

- We are uncovering better ways of developing software by doing it and helping others do it.
- Through this work we have come to value:



- That is, while there is value in the items on the right, we value the items on the left more.

THE 12 PRINCIPLES BEHIND THE AGILE MANIFESTO [1/4]

1. Our highest priority is to satisfy the customer through early and **continuous delivery** of valuable software.
2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter

THE 12 PRINCIPLES BEHIND THE AGILE MANIFESTO [2/4]

4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

THE 12 PRINCIPLES BEHIND THE AGILE MANIFESTO [3/4]

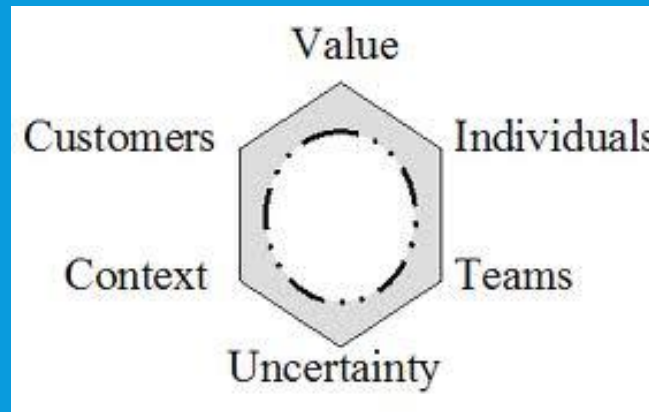
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.

THE 12 PRINCIPLES BEHIND THE AGILE MANIFESTO

10. Simplicity—the art of maximizing the amount of work not done is essential.
11. The best architectures, requirements, and designs emerge from self organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

DECLARATION OF INTERDEPENDENCE

1. Increase **return on investment** by making continuous flow of **value** our focus.
2. Deliver **reliable results** by engaging **customers** in frequent interactions and shared ownership.
3. Expect **uncertainty** and manage for it through iterations, anticipation and adaptation.
4. Unleash **creativity and innovation** by recognizing that **individuals** are the ultimate source of value, and creating an environment where they can make a difference.
5. Boost **performance** through **group accountability** for results and **shared responsibility** for team effectiveness.
6. Improve **effectiveness and reliability** through **situationally specific strategies, processes, and practices**.



KNOWLEDGE WORK VS INDUSTRIAL WORK

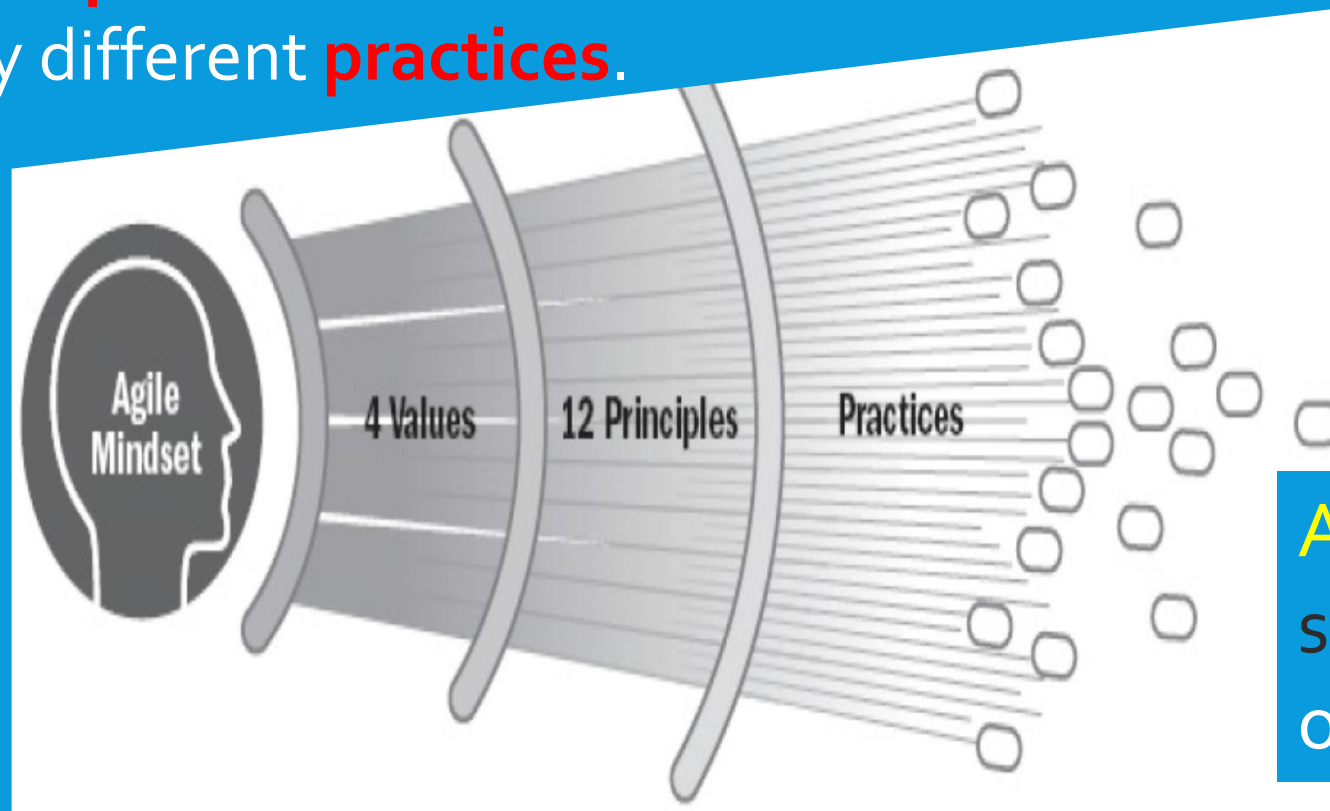


جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

Characteristics of Industrial Work	Characteristics of Knowledge Work
Work is visible	Work is invisible
Work is stable	Work is changing
Emphasis is on running things	Emphasis is on changing things
More structure with fewer decisions	Less structure with more decisions
Focus on the right answers	Focus on the right questions
Define the task	Understand the task
Command and control	Give autonomy
Strict standards	Continuous innovation
Focus on quantity	Focus on quality
Measure performance to strict standards	Continuously learn and teach
Minimize cost of workers for a task	Treat workers as assets, not as costs

THE AGILE MINDSET

✓ Agile is a mindset defined by **values**, guided by **principles** and manifested through many different **practices**.



Agile practitioners select **practices** based on their **needs**.

KANBAN METHOD

- The Kanban Method is inspired by the original lean-manufacturing system and used specifically for knowledge work.
- It emerged in the mid-2000s as an alternative to the agile methods that were prevalent at the time.

KANBAN METHOD

- The Kanban Method is less prescriptive than some agile approaches and less disruptive, as it is the original “start-where-you-are” approach.
- Project teams can begin applying the Kanban Method with relative ease and progress toward other agile approaches if that is what they deem necessary or appropriate.

UNCERTAINTY, RISK, AND LIFE CYCLE SELECTION

APPROACHES FOR MANAGING UNCERTAIN PROJECTS

- These approaches use:
 - Very short feedback loops,
 - Frequent adaptation of process,
 - Reprioritization,
 - Regularly updated plans, and
 - Frequent delivery.

WHEN TO APPLY ITERATIVE, INCREMENTAL AND AGILE APPROACHES?

- These iterative, incremental, and agile approaches work well for projects that involve:
 - New or novel tools, techniques, materials, or application domains.
 - Require research and development;
 - Have high rates of change;
 - Have unclear or unknown requirements, uncertainty, or risk; or
 - Have a final goal that is hard to describe.

UNCERTAINTY AND DISAGREEMENT

- However, iterative and incremental approaches have their limits of applicability.
- When both technology uncertainty and requirements uncertainty are very high (the top right of Figure 2-5), the project moves beyond complex to chaotic.
- In order for the project to become reliably possible, it needs one of the variables (uncertainty or disagreement) to be contained.

CHARACTERISTICS INVOLVING HIGH UNCERTAINTY

1. product specification / suitability and requirements (is the right product being built?),
2. production capability / technical feasibility and performance (can this product be built this way?)
3. process suitability / process and people (is this an effective way for the team to work?)

OVERCOMING UNCERTAINTY

- By building a small increment and then testing and reviewing it, the team can explore uncertainty at a low cost in a short time, reduce risk, and maximize business value delivery.



Everyone's Smart
University

PMG 622

AGILE PROJECT MANAGEMENT

LECTURE 02: LIFE CYCLE SELECTION

دكتور مهندس استشاري ادارة المشروعات

Eng. Dr. Akram Mustafa Suliman Ali,

PhD. AU, BEng. & MSc. AAST, CEng CMarEng MIMarEST, PE, PMP®,

ASQ CSSBB, ISO Lead Auditor, NDT Level II,



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

CHARACTERISTICS OF PROJECT LIFE CYCLES

TYPES OF PM LIFECYCLES

- **Predictive life cycle.** A more traditional approach, with the bulk of planning occurring upfront, then executing in a single pass; a sequential process.
- **Iterative life cycle.** An approach that allows feedback for unfinished work to improve and modify that work.
- **Incremental life cycle.** An approach that provides finished deliverables that the customer may be able to use immediately.
- **Agile life cycle.** An approach that is both iterative and incremental to refine work items and deliver frequently.

NON-AGILE (NON-ADAPTIVE) APPROACHES

- **Predictive**
- Linear
- Waterfall
- Plan-driven
- Serial

CHARACTERISTICS OF FOUR CATEGORIES OF LIFE CYCLES

Characteristics				
Approach	Requirements	Activities	Delivery	Goal
Predictive	Fixed	Performed once for the entire project	Single delivery	Manage cost
Iterative	Dynamic	Repeated until correct	Single delivery	Correctness of solution
Incremental	Dynamic	Performed once for a given increment	Frequent smaller deliveries	Speed
Agile	Dynamic	Repeated until correct	Frequent small deliveries	Customer value via frequent deliveries and feedback

PREDICTIVE LIFE CYCLES

- Take advantage of things that are known and proven.
- This reduced uncertainty and complexity allows teams to segment work into a sequence of predictable groupings.

ITERATIVE LIFE CYCLES

- Allow feedback on partially completed or unfinished work to improve and modify that work

INCREMENTAL LIFE CYCLES

- Provide finished deliverables that the customer may be able to use immediately

AGILE LIFE CYCLES

- Leverage both the aspects of iterative and incremental characteristics.
- When teams use agile approaches, they iterate over the product to create finished deliverables.
- The team gains early feedback and provides customer visibility, confidence, and control of the product.
- Because the team can release earlier, the project may provide an earlier return on investment because the team delivers the highest value work first.

CHARACTERISTICS OF PREDICTIVE LIFE CYCLES

- Predictive life cycles expect to take advantage of high certainty around firm requirements, a stable team, and low risk
- Project activities often execute in a serial manner
- These projects succeed when potential changes are restricted (plans, team members, requirements)
- Articulate the constraints
- Serialized sequence of work
- Do not typically deliver business value until the end of the
- project

CHARACTERISTICS OF ITERATIVE LIFE CYCLES

- Iterative life cycles improve the product or result through successive prototypes or proofs of concept
- Iterations help identify and reduce uncertainty in the project
- Projects benefit from iterative life cycles when:
 1. complexity is high
 2. when the project incurs frequent changes
 3. when the scope is subject to differing stakeholders' views of the desired final product

CHARACTERISTICS OF INCREMENTAL LIFE CYCLES

- Optimize work for delivering value to sponsors or customers more often than a single, final product
- Delivery of smaller deliverables
- Speed of delivery
- Customers receive a subset of the overall solution frequently
- The degree of change and variation is less important than ensuring customers get value sooner than at the end of the project.

CHARACTERISTICS OF AGILE LIFE CYCLES

- Agile life cycles are those that fulfill the principles of the Agile Manifesto
- Customer satisfaction increases with early and continuous delivery of valuable products
- An incremental deliverable that is functional and provides value is the primary measure of progress
- Adapt to high degrees of change and deliver project value more often
- Incremental delivery uncovers hidden or misunderstood requirements
- Iteration-Based Agile
- Flow-Based Agile
- The iterative and incremental approaches provide feedback to better plan the next part of the project

CHARACTERISTICS OF ITERATION-BASED AGILE LIFE CYCLES

- The team works in iterations (time-boxes of equal duration) to deliver completed features
- The team works on the most important feature, collaborating as a team to finish it
- Then the team works on the next most important feature and finishes it
- The team may decide to work on a few features at a time, but the team does not address all of the work for the iteration at once

CHARACTERISTICS OF FLOW-BASED AGILE LIFE CYCLES

- In flow-based agile, the team pulls features from the backlog based on its capacity to start work rather than on an iteration-based schedule
- The team defines its workflow with columns on a task board and manages the work in progress for each column
- Each feature may take a different amount of time to finish
- Teams keep work-in-progress sizes small to better identify issues early and reduce rework should changes be required
- Without iterations to define planning and review points, the team and business stakeholders determine the most appropriate schedule for planning, product reviews, and retrospectives

PROJECT FACTORS THAT INFLUENCE TAILORING [1/2]

Project Factor	Tailoring Options
Demand pattern: steady or sporadic	Many teams find that using a cadence (in the form of a regular timebox) helps them demo, retrospect, and take in new work. In addition, some teams need more flexibility in their acceptance of more work. Teams can use flowbased agile with a cadence to get the best of both Worlds
Rate of process improvement required by the level of team experience	Retrospect more often and select improvements
The flow of work is often interrupted by various delays or impediments	Consider making work visible using Kanban boards and experimenting with limits for the various areas of the work process in order to improve flow

PROJECT FACTORS THAT INFLUENCE TAILORING [2/ 2]

Project Factor	Tailoring Options
The quality of the product increments is poor	Consider using the various test-driven development practices. This mistake-proofing discipline makes it difficult for defects to remain Undetected
More than one team is needed to build a product	To scale from one to several agile teams, with minimal disruption, first learn about agile program management or formal scaling frameworks. Then, craft an approach that fits the project context
The project team members are inexperienced in the use of agile approaches	Consider starting by training team members in the fundamentals of the agile mindset and principles. If the team decides to use a specific approach such as Scrum or Kanban, provide a workshop on that approach so the team members can learn how to use it



Everyone's Smart
University

PMG 622

AGILE PROJECT MANAGEMENT

LECTURE 03: IMPLEMENTING AGILE: CREATING AN AGILE ENVIRONMENT

دكتور مهندس استشاري ادارة المشروعات

Eng. Dr. Akram Mustafa Suliman Ali,

PhD. AU, BEng. & MSc. AAST, CEng CMarEng MIMarEST, PE, PMP®,

ASQ CSSBB, ISO Lead Auditor, NDT Level II,



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

CREATING AN AGILE ENVIRONMENT

START WITH AN AGILE MINDSET

- How can the project team act in an agile manner?
- What can the team deliver quickly and obtain early feedback to benefit the next delivery cycle?
- How can the team act in a transparent manner?
- What work can be avoided in order to focus on high-priority items?
- How can a servant-leadership approach benefit the achievement of the team's goals?



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

SERVANT LEADERSHIP EMPOWERS THE TEAM

SERVANT LEADERSHIP

- Servant leadership is the practice of leading through service to the team, by focusing on understanding and addressing the needs and development of team members in order to enable the highest possible team performance
- The role of a servant leader is to facilitate the team's discovery and definition of agile.
- Servant leaders practice and radiate agile
- Servant leadership = facilitative skills

SERVANT LEADERS APPROACH PROJECT WORK IN THIS ORDER:

- **Purpose.** Work with the team to define the “why” or purpose so they can engage and coalesce around the goal for the project. The entire team optimizes at the project level, not the person level.
- **People.** Once the purpose is established, encourage the team to create an environment where everyone can succeed. Ask each team member to contribute across the project work.
- **Process.** Do not plan on following the “perfect” agile process, but instead look for the results. When a **cross-functional team delivers finished value often** and **reflects on the product and process**, the teams are agile. It does not matter what the team calls its process

INTERPERSONAL SKILLS VERSUS TECHNICAL SKILLS

- Team members emphasize their interpersonal and emotional intelligence skills—not just technical skills.
- Everyone on the team works to exhibit more initiative, integrity, emotional intelligence, honesty, collaboration, humility, and willingness to communicate in various ways so that the entire team can work together well.
- The team needs these skills so they can respond well to changes in project direction and technical product changes. When everyone can adapt to the work and to each other, the entire team is more likely to succeed

SERVANT LEADERSHIP: ENABLES PROJECT LEADERS TO BECOME MORE AGILE

- Promoting self-awareness;
- Listening;
- Serving those on the team;
- Helping people grow;
- Coaching vs. controlling;
- Promoting **safety**, respect, and trust; and
- Promoting the energy and intelligence of others

SERVANT LEADER RESPONSIBILITIES

- Manage relationships to build communication and coordination within the team and across the organization
- These relationships help the leaders navigate the organization to support the team
- This kind of support helps to remove impediments and facilitates the team to streamline its processes
- Because servant leaders understand agile and practice a specific approach to agile, they can assist in fulfilling the team's needs

SERVANT LEADERS PAVE THE WAY FOR OTHERS' CONTRIBUTION

- In agile, the team manages its work process and its work product.
- Servant leaders work to fulfill the needs of the teams, projects, and organization.
- The servant leader focuses on paving the way for the team to do its best work.
- The servant leader influences projects and encourages the organization to think differently.

EXAMPLES OF THE RESPONSIBILITIES A SERVANT LEADER MAY HAVE:

- Educate stakeholders around why and how to be agile
- Support the team through mentoring, encouragement, and support.
- Advocate for team members training and career development.
- Help the team with technical project management activities like quantitative risk analysis.
- Celebrate team successes and support and bridge building activities with external groups.
- Create upward spirals of appreciation and good will for increased collaboration.

ROLE OF THE PROJECT MANAGER IN AN AGILE ENVIRONMENT

- Some agile practitioners think the role of a project manager is not needed, due to self-organizing teams taking on the former responsibilities of the project manager.
- However, pragmatic agile practitioners and organizations realize that project managers can add significant value in many situations. The key difference is that their roles and responsibilities look somewhat different.

PROJECT MANAGERS USE SERVANT LEADERSHIP

- When working on an agile project, project managers shift from being the center to serving the team and the management.
- In an agile environment, project managers are servant leaders, changing their emphasis to coaching people who want help, fostering greater collaboration on the team, and aligning stakeholder needs.
- As a servant leader, project managers encourage the distribution of responsibility to the team: to those people who have the knowledge to get work done.

ATTRIBUTES OF SUCCESSFUL AGILE TEAMS

[1/2]

Attribute	Goal
Dedicated people	• Increased focus and productivity • Small team, fewer than ten people
Cross-functional team members	• Develop and deliver often • Deliver finished value as an independent team • Integrate all the work activities to deliver finished work • Provide feedback from inside the team and from others, such as the product owner
Colocation or ability to manage any location challenges	• Better communication • Improved team dynamics • Knowledge sharing • Reduced cost of learning • Able to commit to working with each other

ATTRIBUTES OF SUCCESSFUL AGILE TEAMS

[2/2]

Attribute	Goal
Mixed team of generalists and specialists	• Specialists provide dedicated expertise and generalists provide flexibility of who does what • Team brings their specialist capabilities and often become generalizing specialists, with a focus specialty plus breadth of experience across multiple skills
Stable work environment	• Depend on each other to deliver • Agreed-upon approach to the work • Simplified team cost calculations (run rate) • Preservation and expansion of intellectual capital

AGILE ROLES

- Cross-functional team members,
- Product owner, and
- Team facilitator.

CROSS-FUNCTIONAL TEAM MEMBER

- Cross-functional teams consist of team members with all the skills necessary to produce a working product.
- In software development, cross-functional teams are typically comprised of designers, developers, testers, and any other required roles.
- The cross-functional development teams consist of professionals who deliver potentially releasable product on a regular cadence.
- Cross-functional teams are critical because they can deliver finished work in the shortest possible time, with higher quality, without external dependencies.

PRODUCT OWNER [1/3]

- The product owner is responsible for guiding the direction of the product.
- Product owners rank the work based on its business value.
- Product owners work with their teams daily by providing product feedback and setting direction on the next piece of functionality to be developed/delivered.

PRODUCT OWNER [2/3]

- The product owner works with stakeholders, customers, and the teams to define the product direction.
- Typically, product owners have a business background and bring deep subject matter expertise to the decisions.
- Sometimes, the product owner requests help from people with deep domain expertise, such as architects, or deep customer expertise, such as product managers.
- Product owners need training on how to organize and manage the flow of work through the team.

PRODUCT OWNER [3/3]

- In agile, the product owners create the backlog for and with the team.
- The backlog helps the teams see how to deliver the highest value without creating waste.
- A critical success factor for agile teams is strong product ownership.
- Without attention to the highest value for the customer, the agile team may create features that are not appreciated, or otherwise insufficiently valuable, therefore wasting effort.
- *That means the work is small, often small enough to be described on one index card.

TEAM FACILITATOR

- A servant leader.
- This role may be called a project manager, scrum master, project team lead, team coach, or team facilitator.
- People need time to build their **servant leadership skills** of facilitation, coaching, and impediment removal.
- Initially, many organizations invite external agile coaches to help them when their internal coaching capability is not yet fully developed.
- External coaches have the advantage of experience, but the disadvantage of weak relationships in the client organization.
- Internal coaches, on the other hand, have strong relationships in their organization, but may lack the breadth of experience that would make them highly effective.



Everyone's Smart
University

PMG 622

AGILE PROJECT MANAGEMENT

LECTURE 04: IMPLEMENTING AGILE: DELIVERING IN AN AGILE ENVIRONMENT **[PART 1 OF 2]**

دكتور مهندس استشاري ادارة المشروعات

Eng. Dr. Akram Mustafa Suliman Ali,

PhD. AU, BEng. & MSc. AAST, CEng CMarEng MIMarEST, PE, PMP®,

ASQ CSSBB, ISO Lead Auditor, NDT Level II,



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

DELIVERING IN AN AGILE ENVIRONMENT

CHARTER THE PROJECT AND THE TEAM

- Project Charter
- Team Charter

PROJECT CHARTER

- Why this project matters **The WHY**
- Where the team is headed **The HOW**
- What the project objective is **The WHAT**
- Why are we doing this project? This is the project vision.
- Who benefits and how? This may be part of the project vision and/or project purpose.
- What does done mean for the project? These are the project's release criteria.
- How are we going to work together? This explains the intended flow of work.

TEAM CHARTER

- Team norms
- Understanding of how to work together
- Teams do not need a formal process for chartering as long as the teams understand how to work together.
- Some teams benefit from a team chartering process.
 - ☐ Team values, such as sustainable pace and core hours;
 - ☐ Working agreements, such as what “**ready**” means so the team can take in work; what “**done**” means so the team can judge completeness consistently; respecting the **time-box**; or the use of **work-in-process** limits;
 - ☐ Ground rules, such as one person talking in a meeting; and
 - ☐ Group norms, such as how the team treats meeting times.

COMMON AGILE PRACTICES

RETROSPECTIVES [1/2]

- The single most important practice is the retrospective because it allows the team to learn about, improve, and adapt its process
- Retrospectives help the team learn from its previous work on the product and its process
- One of the principles behind the **Agile Manifesto** is: “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”
- Teams benefit from allocating enough time to learn, either from an **interim retrospective** or an **end-of-the-project retrospective**

RETROSPECTIVES [2/2]

- Teams need to learn about their **work product** and **work process**.
- First and foremost, a retrospective is **not** about **blame**;
- The retrospective is about looking at the **qualitative (people's feelings)** and
- **quantitative (measurements) data**, then using that data to find root causes, designing countermeasures, and developing action plans.
- The project team may end up with many action items to remove **impediments**
- A facilitator from the team leads them through an activity to rank the importance of each improvement item.
- Once the improvement items are ranked by the team, the team chooses the appropriate number to work on for the next iteration (or adds work to the flow if flow-based).

BACKLOG PREPARATION

- The backlog is the **ordered list of all the work**, presented in **story** form, for a team.
- There is no need to create all of the stories for the entire project before the work starts—only enough to understand the first release in broad brushstrokes and then sufficient items for the next iteration.
- **Product owners** might produce a **product roadmap** to show the anticipated sequence of deliverables over time. The product owner re-plans the roadmap based on what the team produces.

BACKLOG REFINEMENT

- In **iteration-based agile**, the **product owner** often works with the team to prepare some **stories** for the upcoming iteration during one or more sessions in the middle of the iteration.
- The **purpose** of these meetings is to **refine** enough **stories** so the **team** understands what the stories are and **how large** the stories are in relation to each other.
- Allow the **product owner** to **present story ideas** to the **team** and for the team to learn about the potential challenges or problems in the stories.
- If the **product owner** is **unsure** of the **dependencies**, the product owner can request the **team** to **spike** the feature in order to understand the risks.

HOW LONG THE REFINEMENT SHOULD BE

- Many **iteration-based agile** teams use a **time-boxed** 1-hour discussion midway through a 2-week iteration.
- **Just-in-time** refinement for **flow-based agile**. The team takes the next card off the to-do column and discusses it.
- **Multiple refinement discussions** for **iteration-based agile teams**. Teams can use this when they are **new to the product**, the **product area**, or the **problem domain**.

DAILY STANDUPS

- Teams use standups to **micro-commit** to each other, **uncover problems**, and **ensure the work flows smoothly** through the team.
- **Time-box** the standup to no longer than **15 minutes**. The team “walks” the **Kanban** or **task board** in some way, and **anyone** from the **team can facilitate** the standup.

ITERATION-BASED AGILE DAILY STAND-UP QUESTIONS

- What did I complete since the last standup?
- What am I planning to complete between now and the next standup?
- What are my impediments (or risks or problems)?

FLOW-BASED AGILE DAILY STAND-UP QUESTIONS

- What do we need to do to advance this piece of work?
- Is anyone working on anything that is not on the board?
- What do we need to finish as a team?
- Are there any bottlenecks or blockers to the flow of work?

STANDUP ANTIPATTERNS

- Standups become status meetings. Teams who have traditionally worked in a predictive environment may tend to fall into this antipattern since they are used to providing a status.
- The team begins to solve problems as they become apparent. Standups are for realizing there are problems —not for solving them. Add the issues to a parking lot, and then create another meeting, which might be right after the standup, and solve problems there.
- Standups can be very useful, provided the nature of the team's work requires intense collaboration.

DEMONSTRATIONS/REVIEWS

- As the team completes the features usually in the form of user stories, the team periodically demonstrates the working product.
- The product owner sees the demonstration and accepts or declines stories.
- In iteration-based agile, the team demonstrates all completed work items at the end of the iteration.
- In flow-based agile, the team demonstrates completed work when it is time to do so, usually when enough features have accumulated into a set that is coherent.

GUIDELINES

- A fundamental part of what makes a project agile is the frequent delivery of a working product.
- Teams, including the product owner, need feedback to decide how early to ask for product feedback
- Demonstrate whatever the team has as a working product at least once every 2 weeks

HOW ITERATIONS AND INCREMENTS HELP DELIVER WORKING PRODUCT

- Iterations help a team create a cadence for delivery and many kinds of feedback.
- Teams produce increments of value for delivery and feedback.
- The first part of this delivery is a demonstration.
- Teams receive feedback about how the product looks and operates through a demo.
- Team members retrospect to see how they can inspect and adapt their process to succeed.
- Demonstrations or reviews are a necessary part of the agile project flow.
- Schedule the demonstration as appropriate for the team's delivery cadence.



Everyone's Smart
University

PMG 622

AGILE PROJECT MANAGEMENT

LECTURE 05: IMPLEMENTING AGILE: DELIVERING IN AN AGILE ENVIRONMENT [PART 2 OF 2]

دكتور مهندس استشاري ادارة المشروعات

Eng. Dr. Akram Mustafa Suliman Ali,

PhD. AU, BEng. & MSc. AAST, CEng CMarEng MIMarEST, PE, PMP®,

ASQ CSSBB, ISO Lead Auditor, NDT Level II,



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

TROUBLESHOOTING AGILE PROJECT CHALLENGES

AGILE PAIN POINTS AND TROUBLESHOOTING POSSIBILITIES [1/8]

Pain Point	Troubleshooting Possibilities
Unclear purpose or mission for the team	Agile chartering for purpose–vision, mission, and mission tests
Unclear working agreements for the team	Agile chartering for alignment–values, principles, and working agreements
Unclear team context	Agile chartering for context–boundaries, committed assets, and prospective analysis

AGILE PAIN POINTS AND TROUBLESHOOTING POSSIBILITIES [2/8]

Pain Point	Troubleshooting Possibilities
Unclear requirements	❖ Help sponsors and stakeholders craft a product vision. ❖ Consider building a product roadmap using specification by example, user story mapping, and impact mapping. ❖ Bring the team and product owner together to clarify the expectations and value of a requirement. ❖ Progressively decompose roadmap into backlog of smaller, concrete requirements.
Poor user experience	User experience design practices included in the development team involve users early and often.

AGILE PAIN POINTS AND TROUBLESHOOTING POSSIBILITIES [3/8]

Pain Point	Troubleshooting Possibilities
Inaccurate estimation	❖ Reduce story size by splitting stories. ❖ Use relative estimation with the entire team to estimate. ❖ Consider agile modeling or spiking to understand what the story is.
Unclear work assignments or work progress	❖ Help the team learn that they self-manage their work. ❖ Consider kanban boards to see the flow of work. ❖ Consider a daily standup to walk the board and see what work is where.

AGILE PAIN POINTS AND TROUBLESHOOTING POSSIBILITIES [4/8]

Pain Point	Troubleshooting Possibilities
Team struggles with obstacles	❖ A servant leader can help clear these obstacles. ❖ If the team doesn't know the options they have, consider a coach. ❖ Sometimes, the team needs to escalate stories the team or servant leader has not been able to remove.
Work delays/overruns due to Insufficiently refined product backlog items	❖ Product owner and team workshop stories together. ❖ Create a definition of ready for the stories. ❖ Consider splitting stories to use smaller stories
Defects	❖ Consider the technical practices that work for the environment. ❖ Some possibilities are: pair work, collective product ownership, pervasive testing (test-driven and automated testing approaches) and a robust definition of done.

AGILE PAIN POINTS AND TROUBLESHOOTING POSSIBILITIES [5/8]

Pain Point	Troubleshooting Possibilities
Work is not complete	❖ Team defines definition of done for stories including acceptance criteria. ❖ Also add release criteria for projects.
Technical debt (degraded code quality)	Refactoring, agile modeling, pervasive testing, automated code quality analysis, definition of Done
Slow or no improvement in the teamwork process	❖ Capture no more than three items to improve at each retrospective. ❖ Ask the servant leader to help the team learn how to integrate those items.

AGILE PAIN POINTS AND TROUBLESHOOTING POSSIBILITIES [6/8]

Pain Point	Troubleshooting Possibilities
Too much product complexity	For software and non-software encourage the team always to be thinking “What is the simplest thing that would work?” and apply the agile principle of <u>“Simplicity--the art of maximizing the amount of work not done”</u> . These help reduce complexity.
Too much upfront work leading to rework	❖ Instead of much upfront work, consider team spikes to learn. ❖ In addition, measure the WIP during the beginning of the project and see what the team's options are to deliver value instead of designs. ❖ Shorten iterations and create a robust definition of done.

AGILE PAIN POINTS AND TROUBLESHOOTING POSSIBILITIES [7/8]

Pain Point	Troubleshooting Possibilities
False starts, wasted efforts	Ask the product owner to become an integral part of the team.
Inefficiently ordered product backlog items	Rank with value including cost of delay divided by duration and other value models
Rush/wait uneven flow of work	❖ Plan to the team's capacity and not more. ❖ Ask people to stop multitasking and be dedicated to one team. ❖ Ask the team to work as pairs, a swarm, or mob to even out the capabilities across the entire team.

AGILE PAIN POINTS AND TROUBLESHOOTING POSSIBILITIES [8/8]

Pain Point	Troubleshooting Possibilities
Impossible stakeholder demands	Servant leadership to work with this stakeholder (and possibly product owner).
Unexpected or unforeseen delays	❖ Ask the team to check in more often, use Kanban boards to see the flow of work and work in progress limits to understand the impact of the demands on the team or product. ❖ Also track impediments and impediment removal on an impediment board.
Siloed teams, instead of cross-functional teams	❖ Ask the people who are part of projects to selforganize as cross-functional teams. ❖ Use servant leadership skills to help the managers understand why agile needs cross-functional teams.



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

MEASUREMENTS IN AGILE PROJECTS

AGILE TEAMS MEASURE RESULTS

- Agile favors empirical and value-based measurements instead of predictive measurements.
- Agile measures what the team delivers, not what the team predicts it will deliver.
- Agile is based on working products of demonstrable value to customers and not managing to a baseline

AGILE DURATION PREDICTION

- Once the team establishes a reliable velocity (average stories or story points per iteration) or the average cycle time, the team can predict how much longer the project will take
- Teams might discover it can take four to eight iterations to achieve a stable velocity. The teams need the feedback from each iteration to learn about how they work and how to improve.

VELOCITY,

- The sum of the story point sizes for the features actually completed in this iteration, allows the team to plan its next capacity more accurately by looking at its historical performance.

THE DOWNSIDE OF RELATIVE ESTIMATION

- There is no way to compare teams or add velocity across teams



Everyone's Smart
University

PMG 622

AGILE PROJECT MANAGEMENT

LECTURE 06: ORGANIZATIONAL CONSIDERATIONS FOR PROJECT AGILITY

دكتور مهندس استشاري ادارة المشروعات

Eng. Dr. Akram Mustafa Suliman Ali,

PhD. AU, BEng. & MSc. AAST, CEng CMarEng MIMarEST, PE, PMP®,

ASQ CSSBB, ISO Lead Auditor, NDT Level II,

INTRODUCTION

- ❑ Cultures, structures, and policies can influence both the direction and the outcome of any project.
- ❑ Project leaders are expected to navigate organizational dynamics skillfully
- ❑ Leaders can explore options for change, to increase project success
- ❑ Project agility is more effective and sustained as the organization adjusts to support it

ORGANIZATIONAL CHANGE MANAGEMENT

INTRODUCTION

- ❑ Organizational change management covers the skills and techniques for influencing changes that support agility

DRIVERS FOR CHANGE MANAGEMENT

- ❑ All projects are about change. However, there are two key factors that further motivate the use of change management practices in an agile context:
 - ❑ **Changes associated with accelerated delivery.**
 - ❑ Agile approaches emphasize delivering project outputs early and often.
 - ❑ However, the receiving organization may not be fully prepared to incorporate those outputs at an increased pace.
 - ❑ Customer acceptance of and alignment with project outputs becomes even more prevalent in an agile environment.
 - ❑ **Changes associated with agile approaches.**
 - ❑ Higher degrees of collaboration may require more frequent handoffs between teams, departments, or vendors.
 - ❑ Decomposing work into iterative prototypes involves rework that could be viewed negatively.
 - ❑ Leaders should consider change management techniques to address the hurdles of transitioning to the use of agile approaches.



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

READINESS FOR CHANGE

ORGANIZATIONAL CHANGE-FRIENDLY CHARACTERISTICS

- ☐ Executive management's willingness to change;
- ☐ Organization's willingness to shift the way it views, reviews, and assesses employees;
- ☐ Centralization or decentralization of project, program, and portfolio management functions;
- ☐ Focus on short-term budgeting and metrics versus long-term goals;
- ☐ Talent management maturity and capabilities.

ORGANIZATIONAL ROADBLOCK TO CHANGE

- ☐ Work is decomposed into departmental silos, creating dependencies that prevent accelerated delivery instead of building cross-functional teams with guidance from centers of competencies.
- ☐ Procurement strategies are based on short-term pricing strategies, rather than long-term competencies.
- ☐ Leaders are rewarded for local efficiencies rather than end-to-end flow of project delivery or optimizing the whole (in regard to the organization).
- ☐ Employees are specialized contributors with limited tools or incentives to diversify their skills instead of building T-shaped specialists.
- ☐ Decentralized portfolios pull employees simultaneously onto too many projects at once instead of keeping them focused on one project at a time.



Everyone's Smart
University

PMG 622

AGILE PROJECT MANAGEMENT

LECTURE 07: ANNEX A3

OVERVIEW OF AGILE AND LEAN

FRAMEWORKS [1/2]

دكتور مهندس استشاري ادارة المشروعات

Eng. Dr. Akram Mustafa Suliman Ali,

PhD. AU, BEng. & MSc. AAST, CEng CMarEng MIMarEST, PE, PMP®,

ASQ CSSBB, ISO Lead Auditor, NDT Level II,

INTRODUCTION TO AGILE AND LEAN FRAMEWORKS

- These approaches can be used as is or combined to adapt to what works best for a given environment or situation.
- It is **NOT** necessary to use any of these; an agile approach can be developed from scratch as long as it adheres to the mindset, values, and principles of the Agile Manifesto.
- If the agile principles are followed to deliver value at a sustainable pace, and the developed approach promotes collaboration with the customer, a specific approach is not required.

SELECTION CRITERIA FOR THE *AGILE PRACTICE* *GUIDE*

- **Designed for holistic use**
 - Intended to guide a broad set of project activities
- **Formalized for common use**
 - Intended for common use in a variety of contexts
- **Popular in modern use**
 - Adopted by a significant number of industries



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

SCRUM

SCRUM

- Scrum is a single-team process framework used to manage product development.
- The framework consists of Scrum roles, events, artifacts, and rules, and uses an iterative approach to deliver working product.
- Scrum is run on timeboxes of 1 month or less with consistent durations called sprints where a potentially releasable increment of product is produced.

THE SCRUM TEAM (ROLES)

- The product owner is responsible for maximizing the value of the product.
- The development team is a cross-functional, self-organizing team consisting of team members who have everything they need within the team to deliver working product without depending on others outside of the team.
- The scrum master is responsible for ensuring the Scrum process is upheld and works to ensure the Scrum team adheres to the practices and rules as well as coaches the team on removing impediments.

SCRUM EVENTS AND ARTIFACTS

Events	Artifacts
Sprint	Product backlog
Sprint planning	Sprint backlog
Daily scrum	Increments
Sprint review	
Sprint retrospective	



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

KANBAN METHOD

KANBAN METHOD [1/2]

- Kanban in lean manufacturing is a system for scheduling inventory control and replenishment.
- This process of “just-in-time” inventory replenishment was originally seen in grocery stores when shelves were restocked based on the gaps in the shelves and not supplier inventory.
- Inspired by these just-in-time inventory systems, Taiichi Ohno developed Kanban and it was applied at the main Toyota manufacturing facility in 1953.
- The Kanban Method is a holistic framework for incremental, evolutionary process and systems change for organizations.

KANBAN METHOD [2/2]

- The method uses a “pull system” to move the work through the process. When the team completes an item, the team can pull an item into that step.
- In the Kanban Method, it is more important to complete work than it is to start new work.
- There is no value derived from work that is not completed so the team works together to implement and adhere to the work in progress (WIP) limits and get each piece of work through the system to “done.”

KANBAN

- The word *kanban* is literally translated as “visual sign” or “card.”
- Physical kanban boards with cards enable and promote the visualization and flow of the work through the system for everyone to see.
- This information radiator (large display) is made up of columns that represent the states the work needs to flow through in order to get to done.
- The simplest of boards could have three columns (i.e., to do, doing, and done), but it is adaptable to whatever states are deemed needed by the team utilizing it.

APPLICABILITY AND UTILIZATION

- The Kanban Method is utilized and applicable in many settings and allows for a continuous flow of work and value to the customer.
- The Kanban Method is less prescriptive than some agile approaches and thus less disruptive to begin implementing as it is the original “start where you are” method.
- Organizations can begin applying Kanban Methods with relative ease and progress toward fully implementing the method if that is what they deem necessary or appropriate.

EMPHASIS

- Unlike most agile approaches, the Kanban Method does not prescribe the use of timeboxed iterations.
- Iterations can be used within the Kanban Method, but the principle of pulling single items through the process continuously and limiting work in progress to optimize flow should always remain intact.

CONDITIONS FOR BEST USE [1/2]

- **Flexibility.** Teams are typically not bound by timeboxes and will work on the highest priority item in the backlog of work.
- **Focus on continuous delivery.** Teams are focused on flowing work through the system to completion and not beginning new work until work in progress is completed.
- **Increased productivity and quality.** Productivity and quality are increased by limiting work in progress.
- **Increased efficiency.** Checking each task for value adding or nonvalue-added activities and removing the non-value adding activities.

CONDITIONS FOR BEST USE [2/2]

- **Team member focus.** Limited work in progress allows the team to focus on the current work.
- **Variability in the workload.** When there is unpredictability in the way that work arrives, and it becomes impossible for teams to make predictable commitments; even for short periods of time.
- **Reduction of waste.** Transparency makes waste visible so it can be removed.

DEFINING PRINCIPLES AND PROPERTIES OF THE KANBAN METHOD

Defining Principles	Core Properties
Start with current state Agree to pursue incremental, evolutionary change Respect the current process, roles, responsibilities, and titles Encourage acts of leadership at all levels	Visualize the workflow Limit work in progress Manage flow Make process policies explicit Implement feedback loops Improve collaboratively

KANBAN BOARD

- Kanban boards are a low-tech, hightouch technology that may seem overly simplistic at first, but those using them soon realize their power.
- Utilizing policies for entry and exit to columns, as well as constraints such as limiting work in process, kanban boards provide clear insight to workflow, bottlenecks, blockers, and overall status.
- Additionally the board acts as an information radiator to anyone who sees it, providing up-to-date information on the status of the work of the team.



Everyone's Smart
University

PMG 622

AGILE PROJECT MANAGEMENT

LECTURE 08: ANNEX A3

OVERVIEW OF AGILE AND LEAN

FRAMEWORKS [2/2] & TAILORING

دكتور مهندس استشاري ادارة المشروعات

Eng. Dr. Akram Mustafa Suliman Ali,

PhD. AU, BEng. & MSc. AAST, CEng CMarEng MIMarEST, PE, PMP®,

ASQ CSSBB, ISO Lead Auditor, NDT Level II,



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

DISCIPLINED AGILE (DA)

DA

- Disciplined Agile (DA) is a process decision framework that integrates several agile best practices into a comprehensive model.
- DA was designed to offer a balance between those popular methods deemed to be either too narrow in focus (e.g., Scrum) or too prescriptive in detail (e.g., AgileUP).

DA PRINCIPLES

- **People-first.** Enumerating roles and organization elements at various levels.
- **Learning-oriented.** Encouraging collaborative improvement.
- **Full delivery life cycle.** Promoting several fit-for-purpose life cycles.
- **Goal-driven.** Tailoring processes to achieve specific outcomes.
- **Enterprise awareness.** Offering guidance on cross-departmental governance.
- **Scalable.** Covering multiple dimensions of program complexity.

APPENDIX X2

ATTRIBUTES THAT INFLUENCE TAILORING

FIRST SOME CAUTIONS!!!!!!!!!!!!!!

- Tailoring is an advanced topic that should be undertaken by experienced practitioners who have been successful using agile approaches as originally described in multiple environments before they consider tailoring them.
- In other words, gain experience and be successful with one approach before attempting to tailor the approach.

SHU-HA-RI

- Model of skills acquisition describes progression from obeying the rules (Shu 守, means to obey and protect), through consciously moving away from the rules (Ha 破, means to change or digress), and finally through steady practice and improvement finding an individual path (Ri 離, means to separate or leave).
- We need to start and practice at the Shu level before we are ready to move to the Ha level to tailor the process or the Ri level to invent a new custom process.



Everyone's Smart
University

PMG 622

AGILE PROJECT MANAGEMENT

LECTURE 09: AGILE PLANNING

دكتور مهندس استشاري ادارة المشروعات

Eng. Dr. Akram Mustafa Suliman Ali,

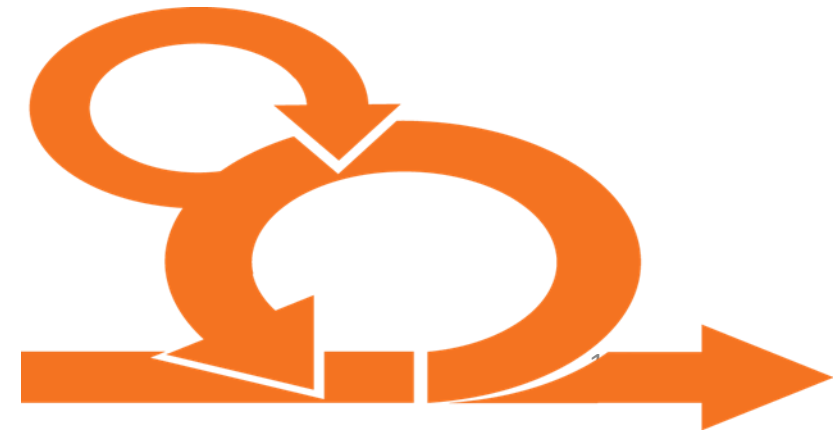
PhD. AU, BEng. & MSc. AAST, CEng CMarEng MIMarEST, PE, PMP®,

ASQ CSSBB, ISO Lead Auditor, NDT Level II,



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

AGILE PLANNING



PLANNING AND TIME BOXING

1. Effectiveness = Planning
2. Efficiency = Time Boxing (Sprints)
3. Failing to plan is planning to fail •

*Planning Is Useful.
Blindly Following the
Plans Is Stupid*

SCRUM' THREE BEST PRACTICES

- Best Practice #1: User Stories
- Best Practice #2: Planning Poker
- Best Practice #3: Use of the Scrum Board

BEST PRACTICE #1: USER STORIES

- Plain-english requirements, written on common 3X5 index cards
- Form: As [a type of user] I want to [perform a specific action] such that [result]
- Example: “As a web user, I want to make a reservation, such that I may secure my lodging”
- Stories that are big are called EPICS
- Acceptance criteria goes on card back

BEST PRACTICE #2: PLANNING POKER

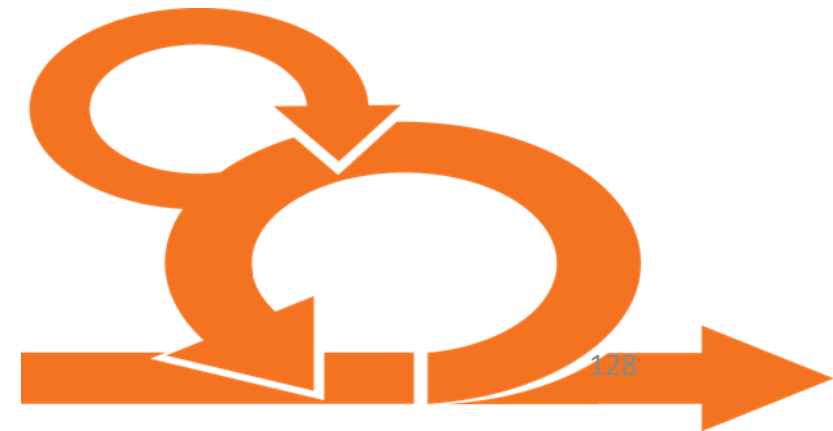
- A way for the team to do estimates
- Each participant has cards numbered 1,2,3,5,8,13,21
- Values represent 'story points' of effort
- Players discuss feature, then throw down a card together
- Differences are noted and discussed, then process repeats till a consensus estimate is formed



COPYRIGHT (c) 2014 FOLIEN.DS. ALL RIGHTS RESERVED. WWW.VECTORART.AT

BEST PRACTICE #3: USE OF THE SCRUM BOARD

- Scrum Board is a rows-and-columns depictions of work-in-progress
- Items of work are rows, work status labels are columns
- Work is addressed from top to bottom
- Work migrates from left to right on the board



ESTIMATING USER STORIES [1/2]

- Agile Estimating uses “Relative Sizing”
 - Since effort is subjective, we use Relative Sizing to other User Stories
- Fibonacci Sequence – 1, 2, 3, 5, 8, 13, 21+
 - We try to keep it under 13 so we can better categorize Stories
- T-Shirt Sizing is popular
 - XS, S, M, L, XL, XXL

ESTIMATING USER STORIES [2/2]

- The number assigned to these are called “Story Points”
 - Story Points can be thought of as “Effort Points” – more points = more effort
- Breakdown of Estimating
 - Epics = Large and Vague
 - Features = A useable product
 - User Story = One part of a feature
 - Task = The smallest item a person could possibly do

AGILE ESTIMATING

- Velocity = The number of Story Points completed during an iteration (Sprint)
 - Points are ONLY added if the User Story is 100% complete
 - Cumulative Average
 - It will take several iterations for Velocity to stabilize
 - Tracked as a TEAM – not for individual performance
 - Velocity will naturally fluctuate (a little) “reasonable amount”
 - The more broken down a Feature can be, the better the estimates will be
 - It can include ANY task so long as there are User Stories (placeholders)



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

AGILE GAMES

FEATURES OF AGILE GAMES

- Designed to promote collaboration
- Have fun
- Gamify the process
- Spark innovation and creativity
- Encourages participation
- Generate organic conversations
- Brings the team closer together
- Promotes diversity

EXAMPLES OF GAMES

- **Planning Poker** – Used to estimate the size of a User Story
- **Coin Game** – Used to show workflow, pull, and batches
- **20/20 Vision** – Used to prioritize User Stories on a board
- **Prune the Product Tree** – Used to prioritize User Stories on a “tree”
- **Product Box** – Used to have stakeholders show what they want and value
- **Buy a Feature** – Used to “buy” features based on priority
- **Draw a House** – Used to encourage reasoning and communication
- **Chair Game** – Used to promote communication and have fun



Everyone's Smart
University

PMG 622

AGILE PROJECT MANAGEMENT

LECTURE 10: ACP

دكتور مهندس استشاري ادارة المشروعات

Eng. Dr. Akram Mustafa Suliman Ali,

PhD. AU, BEng. & MSc. AAST, CEng CMarEng MIMarEST, PE, PMP®,

ASQ CSSBB, ISO Lead Auditor, NDT Level II,

ACP



HOW DOES PMI THINK ABOUT AGILE? [1/3]

- The Agile Coach focuses on clarifying project objectives, coordinating with organizational stakeholders, and assisting the self-managing team
- You [the Coach] are supporting an Agile project team
 - Multi-release schedule
 - Numerous Stakeholders
 - Software project for large organization
- Planning is important, but less important than adapting to change!

HOW DOES PMI THINK ABOUT AGILE? [2/3]

- You support the Agile team throughout the project, key Stakeholders are engaged effectively, roles and responsibilities are defined
- Self-organizing teams own implementation of project processes
- Agile Coaches influence stakeholders, enable change, help to resolve problems, and help teams focus on, and communicate outcomes
- We are always looking for small, incremental improvements

HOW DOES PMI THINK ABOUT AGILE? [3/3]

- Agile Coaches spend their time:
 - Helping the team self-organize and self-manage
 - Helping the team resolve problems and impediments
 - Help engage stakeholders
 - Coach team members and members of the organization on Agile principles
- The Agile approach stipulates:
 - Plan at the high level up-front, and execute releases and iterations, help the team manage work, communicate with stakeholders
 - Planning actions are done by Product Owner working with the team; decisions are made by the team
 - Changes to the plan are sought after, and embraced



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

THE SEVEN AGILE DOMAINS

THE SEVEN AGILE DOMAINS

- Domain 1: Agile Principles & Mindset
- Domain 2: Value Driven Delivery
- Domain 3: Stakeholder Engagement
- Domain 4: Team Performance
- Domain 5: Adaptive Planning
- Domain 6: Problem Detection / Resolution
- Domain 7: Continuous Improvement

DOMAIN 1: AGILE PRINCIPLES & MINDSET

[1/3]

- **TASK 1:** Advocate for agile principles by modeling those principles, and discussing agile values. This will help develop a shared mindset across the team, as well as between the customer and the team.
- **TASK 2:** Help ensure that everyone has a common understanding of the values and principles of agile, and a common knowledge around the agile practices and terminology being used.
- **TASK 3:** Support the change related to embracing agile approaches at the system and organization level by educating individuals, and influencing processes and people.

DOMAIN 1: AGILE PRINCIPLES & MINDSET

[2/3]

- **TASK 4:** Maintain highly visible information radiators, accurately showing progress on the project, and team performance. Use these to enhance transparency and trust.
- **TASK 5:** Contribute to a safe and trusting team environment by allowing everyone to experiment and make mistakes. This enhances learning, and continuous improvement of processes.
- **TASK 6:** Enhance creativity by experimenting with new techniques and process ideas. This will allow the discovery of more effective and efficient ways of working.

DOMAIN 1: AGILE PRINCIPLES & MINDSET

[3/3]

- **TASK 7:** Encourage team members to share knowledge by collaborating and working together. This will help to eliminate knowledge silos, and will reduce bottlenecks in workflows.
- **TASK 8:** Encourage emergent leadership within the team by establishing a safe and respectful environment in which new approaches can be tried. This will foster self-organization, and empower emergent leaders.
- **TASK 9:** Practice servant leadership by supporting and encouraging others. This will help them to perform at their highest level, and continue to improve.

DOMAIN 2: VALUE DRIVEN DELIVERY [1/5]

- **TASK 1:** Define deliverables by identifying Features that can be produced incrementally. This will maximize value to the customer, and minimize work that does not add value.
- **TASK 2:** Refine requirements by gaining consensus on acceptance criteria for Features. These should be focused on value to the customer, and determined in the planning for an Iteration, and before development begins.
- **TASK 3:** Select the processes to be used based on characteristics of the project, and the organization. Tailor the processes based on those same considerations, in whatever way optimizes delivery of value.

DOMAIN 2: VALUE DRIVEN DELIVERY [2/5]

- **TASK 4:** Plan for small releasable increments, focusing on minimally marketable features and minimally viable products. This will allow for early delivery [and recognition] of value.
- **TASK 5:** Limit increment size, and conduct frequent product reviews with stakeholders. This will allow for the early identification of, and response to, variances and risks.
- **TASK 6:** Solicit feedback from the customer and end-users frequently, by reviewing the produced features often. This will help to confirm that the product being delivered is providing real value.

DOMAIN 2: VALUE DRIVEN DELIVERY [3/5]

- **TASK 7:** Collaborate with stakeholders to determine their prioritization of features to be developed, based on value to the customer.
- **TASK 8:** Perform frequent quality checks on the product being developed, and take the necessary actions to maintain the quality of the deliverables. This will reduce the overall cost of the development, and ensure delivery of value.
- **TASK 9:** Continuously identify, and respond to, environmental, operational, and infrastructure factors, to effectively and efficiently produce valuable deliverables.

DOMAIN 2: VALUE DRIVEN DELIVERY [4/5]

- **TASK 10:** Conduct frequent reviews and planning sessions with stakeholders to solicit feedback, and use that feedback to adjust the work to be accomplished next.
- **TASK 11:** Balance actual development / production efforts and risk reduction efforts by including risk reduction work in the Product Backlog. This will optimize the value of the product actually delivered, and overall project success.
- **TASK 12:** Continuously collaborate with the customer and stakeholders to re-prioritize requirements, based on changes to the environment and stakeholder needs.

DOMAIN 2: VALUE DRIVEN DELIVERY [5/5]

- **TASK 13:** Elicit and prioritize relevant non-functional requirements [on-going operations, security, etc.], considering the environment in which the final product will be used. This will help ensure long-term value to the customer.
- **TASK 14:** Conduct frequent inspections, reviews, and testing of the work products. This will help to identify opportunities to improve both the product, and the processes used to produce it.

MY LINKEDIN PAGE

- <https://www.linkedin.com/in/doctorakramali>