Software Design & Architecture

Engr. Abdul-Rahman Mahmood

DPM, MCP, QMR(ISO9001:2000)

armahmood786@yahoo.com
alphapeeler.sf.net/pubkeys/pkey.htm
pk.linkedin.com/in/armahmood
www.twitter.com/alphapeeler
www.facebook.com/alphapeeler
abdulmahmood-sss alphasecure
armahmood786@hotmail.com
http://alphapeeler.sf.net/me

alphasecure@gmail.com
http://alphapeeler.sourceforge.net
http://alphapeeler.tumblr.com
armahmood786@jabber.org
alphapeeler@aim.com
mahmood_cubix 48660186
alphapeeler@icloud.com
http://alphapeeler.sf.net/acms/

Agile Software Engineering

Agile methods

- Dissatisfaction with the overheads involved in design methods led to the creation of agile methods. These methods:
  - **Focus on the code** rather than the design;
  - Are based on an **iterative approach** to software development;
  - Are **intended to deliver working software** quickly and evolve this quickly to meet changing requirements.
- Agile methods are probably best suited to **small/medium-sized business** systems or PC products.

Principles of agile methods

Principle	Description
Customer involvement	The customer should be closely involved throughout the development process. Their role is provide and prioritise new system requirements and to evaluate the iterations of the system.
Incremental delivery	The software is developed in increments with the customer specifying the requirements to be included in each increment.
People not process	The skills of the development team should be recognised and exploited. The team should be left to develop their own ways of working without prescriptive processes.
Embrace change	Expect the system requirements to change and design the system so that it can accommodate these changes.
Maintain simplicity	Focus on simplicity in both the software being developed and in the development process used. Wherever possible, actively work to eliminate complexity from the system.

Problems with agile methods

- It can be **difficult to keep the interest of customers** who are involved in the process.
- **Team members may be unsuited** to the **intense involvement** that characterizes agile methods.
- **Prioritizing changes can be difficult** where there are **multiple stakeholders**.
- Maintaining **simplicity requires extra work**.
- **Contracts may be a problem** as with other approaches to iterative development.

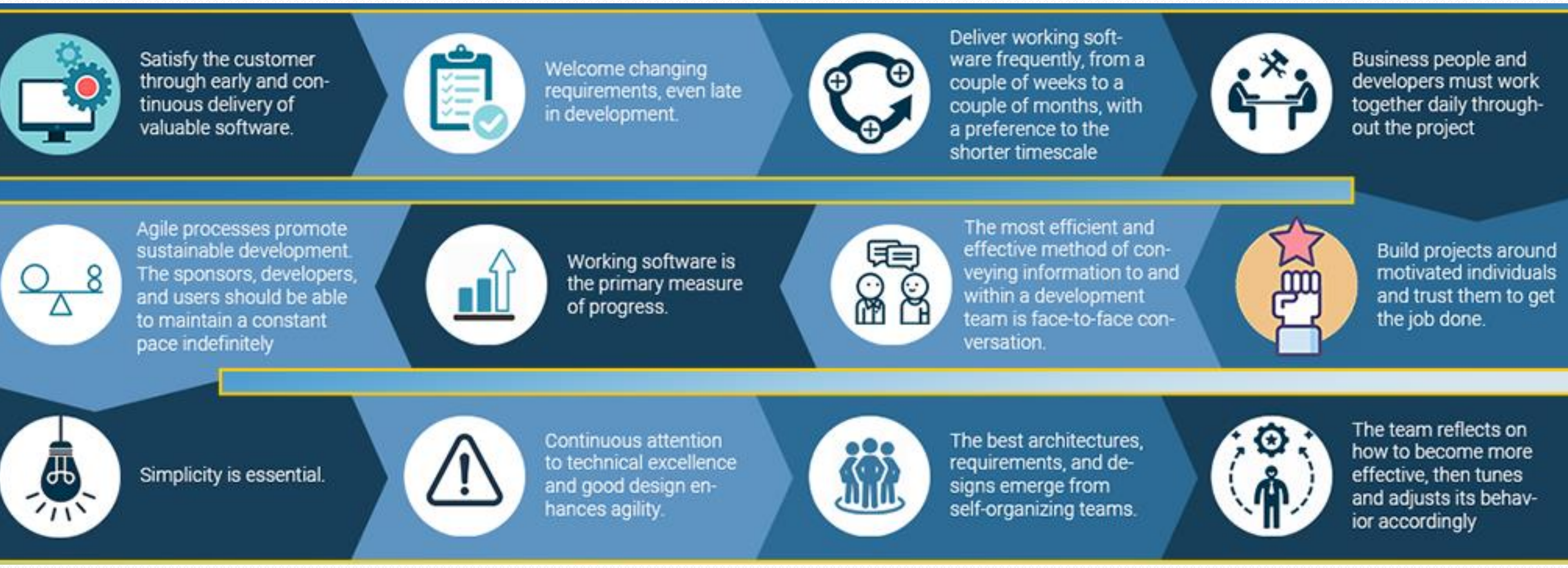
✧ Agile **focus** on small, **tightly-integrated** teams

✧ Problems in **scaling** agile methods to **large systems**.

✧ Less emphasis on **documentation** - harder to maintain

12 Principles of Agility.

- The Alliance created a statement of **values**: termed the **manifesto** of the Agile Alliance.
- They then developed the **12 Principles of Agility.**



Satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale

Business people and developers must work together daily throughout the project

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely

Working software is the primary measure of progress.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Build projects around motivated individuals and trust them to get the job done.

Simplicity is essential.

Continuous attention to technical excellence and good design enhances agility.

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

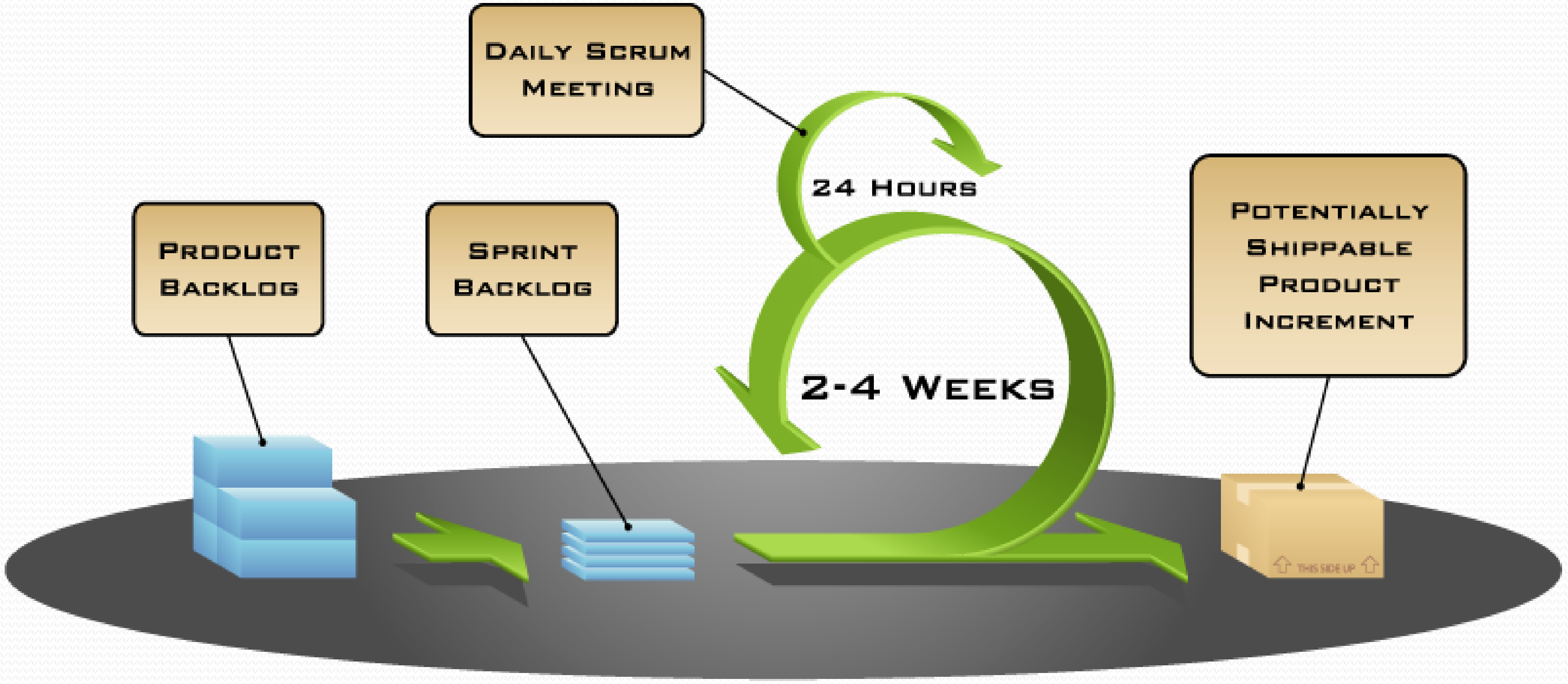
The team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

Examples of Agile Methodology

- The most popular and common examples are Scrum, eXtreme Programming (XP), Feature Driven Development (FDD), Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Crystal, and Lean Software Development (LSD).

  Lets discuss XP and Scrum

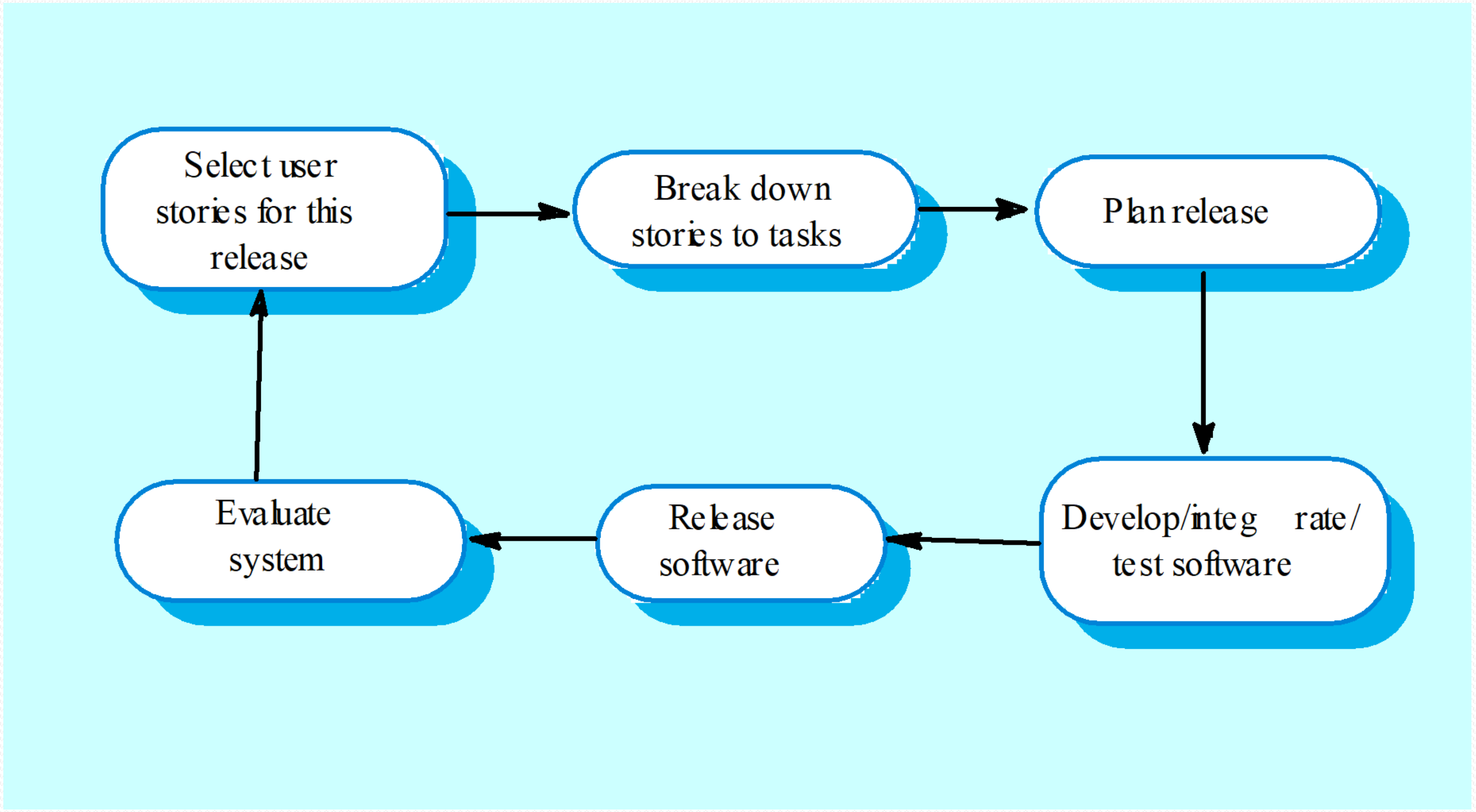
Scrum


Daily Scrum Meeting
24 Hours
Product Backlog
Sprint Backlog
2-4 Weeks
Potentially Shippable Product Increment
This side up

Extreme programming

- The best-known & most **widely used agile method.**
- Extreme Programming (XP) **takes an 'extreme' approach to iterative development.**
  - **New versions** may be built **several times per day**;
  - **Increments** are delivered to customers **every 2 weeks**;
  - All **tests must be run for every build** and the build is only accepted if tests run successfully.

The XP release cycle


Select user stories for this release
Break down stories to tasks
Plan release
Develop/integ rate/ test software
Release software
Evaluate system

Extreme programming practices 1

Incremental planning	Requirements are recorded on Story Cards and the Stories to be included in a release are determined by the time available and their relative priority. The developers break these Stories into development ŌTasksÕ.
Small Releases	The minimal useful set of functionality that provides business value is developed first. Releases of the system are frequent and incrementally add functionality to the first release.
Simple Design	Enough design is carried out to meet the current requirements and no more.
Test first development	An automated unit test framework is used to write tests for a new piece of functionality before that functionality itself is implemented.
Refactoring	All developers are expected to refactor the code continuously as soon as possible code improvements are found. This keeps the code simple and maintainable.

Extreme programming practices 2

Pair Programming	Developers work in pairs, checking ea ch otherÕs work and providing the support to always do a good job.
Collective Ownership	The pairs of developers work on all areas of the system, so that no islands of expertise develop and all the developers own all the code. Anyone can change anything.
Continuous Integration	As soon as work on a task is complete it is integrated into the whole system. After any such integration, all the unit tests in the system must pass.
Sustainable pace	Large amounts of over-time are not considered acceptable as the net effect is often to reduce code qua lity and medium term productivity
On-site Customer	A representative of the end-user of the system (the Customer) should be available full time for the use of the XP team. In an extreme programming process, the customer is a member of the development team and is responsible for bringing system requirements to the team for implementation.

Testing in XP

- Test-first development.
- Incremental test development from scenarios.
- User involvement in test development and validation.
- Automated test harnesses are used to run all component tests each time that a new release is built.

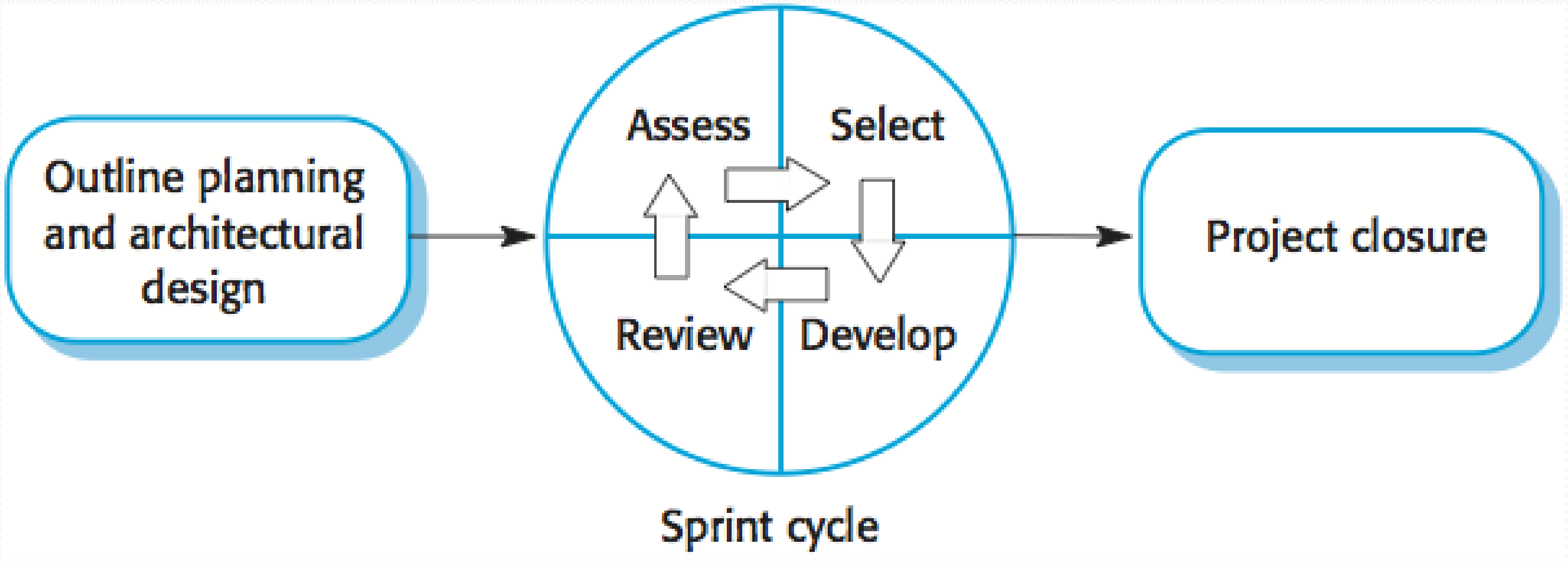
Pair programming

- In XP, programmers work in pairs, sitting together to develop code.
- This helps develop common ownership of code and spreads knowledge across the team.
- It serves as an informal review process as each line of code is looked at by more than 1 person.
- It encourages refactoring as the whole team can benefit from this.
- Measurements suggest that development productivity with pair programming is similar to that of two people working independently.

Scrum

- Project Manager's job: - Deliver needed system on time within budget
- The Scrum approach - manage the iterations
- There are three phases in Scrum.
  - outline <u>planning phase </u>- general picture and architecture
  - <u>Sprint cycles </u>releasing increments of the system.
  - The project <u>closure phase </u>- final delivery, documentation and review of lessons learned.

The Scrum process



The Sprint cycle

- Every 2–4 weeks (a fixed length).
- 1) Project team with customer: Look at product backlog - select stories to implement
- 2) implement with all customer communication through scrum master (protecting pgmr at this point)
  - Scrum master has project manager role during sprint
  - Daily 15 min meetings
    - Stand up often
    - Team presents progress and impediments
    - Scrum master tasked with removing impediments
- 3) Review system release with user

Scrum benefits

- The product is broken down into a set of manageable and understandable chunks.
- Unstable requirements do not hold up progress.
- The whole team have visibility of everything and consequently team communication is improved.
- Customers see on-time delivery of increments and gain feedback on how the product works.
- Trust between customers and developers is established and a positive culture is created in which everyone expects the project to succeed.