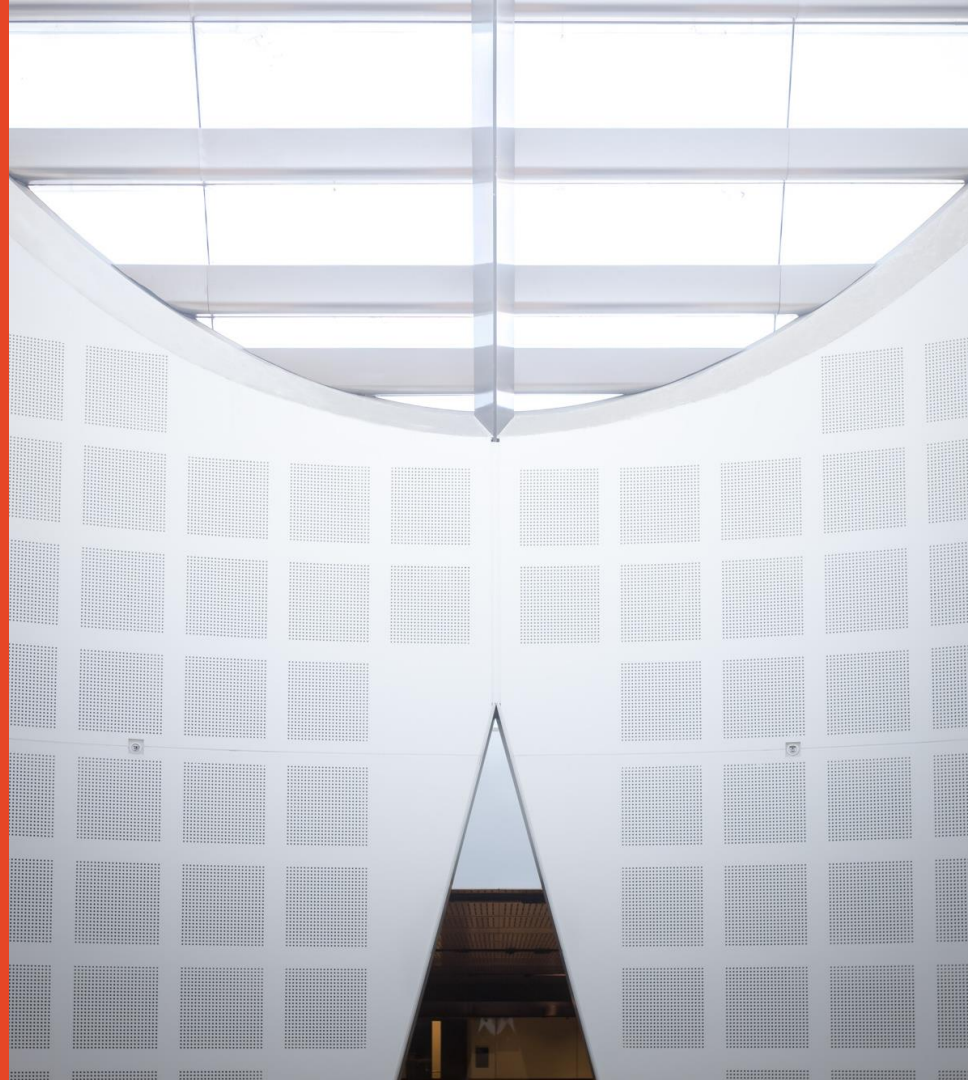


Agile Software Development Practices SOF2412 / COMP9412

Team Dynamics; Tools and
Technologies for Teamwork

Dr. Basem Suleiman

School Computer Science

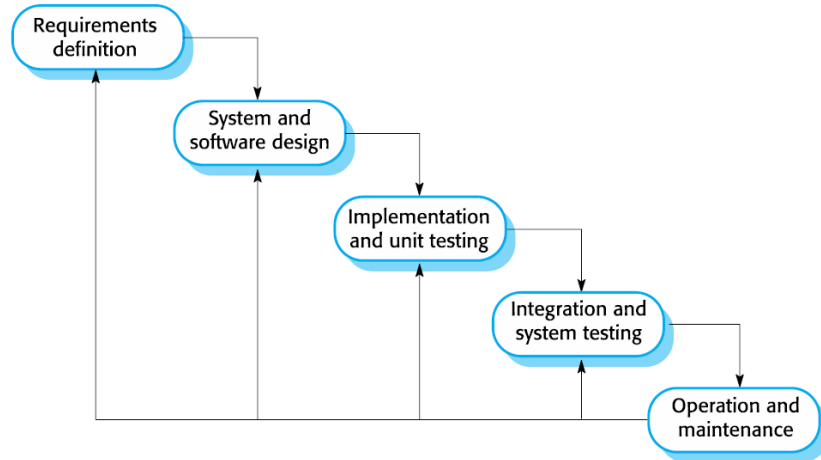


Agenda

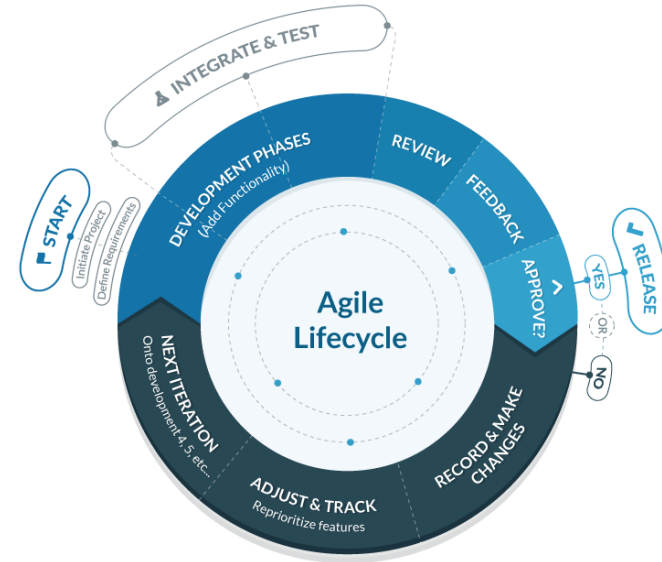
- Teams in Software Development Models
- Teams in Agile Development
- Team Dynamics
- Effective Teams and Teamwork
- Issue Tracking
- Tools for Issue Tracking and Teamwork

Software Development Models – Teams

Do software development models influence team structure and interactions in software development?



Waterfall model
plan-driven development



Agile model
Incremental & iterative development

<https://blog.capterra.com/agile-vs-waterfall/>

Waterfall Model – Teams

| Development activities | Teams |
|--|---|
| Divide the work into stages | A separate team of specialists for each stage |
| At each stage, the work is passed from one team to another | Some coordination is required for the handoff from team to team – using “documents” |
| At the end of all of the stages, you have a software product ready to ship | As each team finishes, they are assigned to a new product |

Teams under different SDLC models

- In a traditional structure how do teams work?
 - As work is planned and allocated, it can be divided into pieces that should be more-or-less independent
 - Specialist teams
 - Project management and resource reallocation
 - Clear authority lines, so disagreements can be resolved
 - Problems?
 - Single points of failure
 - Inflexibility
 - Lack of feedback
- And in Agile teams?

Agile Manifesto – Revisit

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Agile Manifesto –

- Individuals and interactions over processes and tools
- Why Agile values individuals and interactions over processes and tools? Discuss

Agile Manifesto – Why Individuals and Interactions?

- Why Agile values individuals and interactions over processes and tools?
 - People tend to follow processes blindly, and make mistakes
 - “A great tool can sometimes help people to do the wrong thing faster”
 - Tools or best practices are not enough - people who need to use it should buy into it to realize its benefits
 - People needs to see the value of following certain practices
- It is important to recognize that you are working with a group of people who have different motivations, ideas and preferences

Agile Principles 1 – People

- Build projects around **motivated individuals**. Give them the **environment and support** they need, and **trust** them to get the job done
- The most efficient and effective method of **conveying information** to and within a development team is **face-to-face conversation**
- At regular intervals, the **team reflects** on how to become more **effective**, then **tunes and adjusts its behavior** accordingly

Agile Principles 1 – People

- *Group Discussion:*
 - *Briefly discuss how the following agile principles perceive teams and teamwork in agile development?*
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

Agile Principles 2 – People

- Business **people** and **developers** must **work together daily** throughout the project
- The best architectures, requirements, and designs emerge from **self-organizing teams**
- Our highest priority is to **satisfy the customer** through early and continuous delivery of valuable software

Agile Principles 2 – People

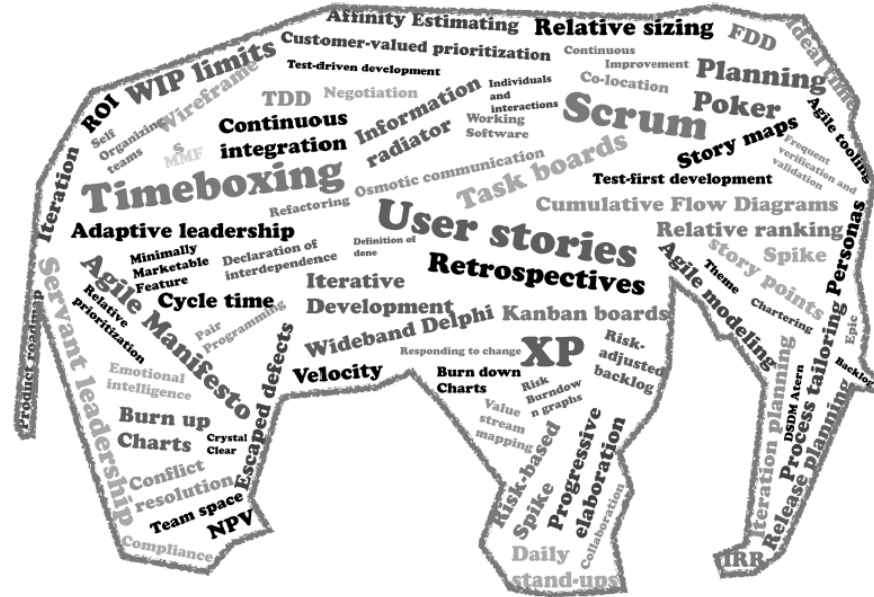
- *Group discussion*
 - *briefly discuss how the following agile principles perceive teams and teamwork in agile development?*
- Business people and developers must work together daily throughout the project
- The best architectures, requirements, and designs emerge from self-organizing teams
- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software

Teams – Individuals and Collaboration

- Common problem experienced in software development teams “*throw it over the wall*”
 - Team members are busy thinking about their own project work and problems
 - Different views/perspectives
 - Teams are divided, and collaboration is killed

Agile Practices – The Agile Elephant

- The agile elephant is made up of many practices



*Agile practices and Scrum method (most of these presented in this figure) will be discussed in more details in future lectures. The focus of the discussion here is on teams and the adoption of Agile practice

Agile Teams – Individual Practices

- When adopting agile practices, team members may adopt practices individually:
 - Each person uses only the practices that impact their work; developers focus on automated tests and build, team leads on task boards, project velocity and burn-down charts, business users on user stories
 - Adopting practices individually will improve things, but this may lead to a self-contradictory effect

Teams – Individual Practices

- When adopting agile practices, team members may follow the same thinking:
 - Each person uses only the practices that impact their work; developers focus on automated tests and build, team leads on task boards, project velocity and burn-down charts, business users on user stories
 - Adopting practices individually will improve things, but this may lead to a self-contradictory effect
 - Each person sees the part of agile that affects their specific work – (attitudes: “see! I was right all along”)
 - Agile is made up of day-to-day practices, but it’s much bigger than those practices

Understanding the Agile Elephant

- If you only see the practices that directly affect your project work, then you will see the one small piece of agile

Not the whole elephant!



Team Dynamics



Team Dynamics

- “Team dynamics are the **unconscious, psychological forces** that influence the direction of a team’s behaviour and performance”
- Factors that lead to team dynamics:
 - Personalities and work styles
 - Knowledge and skills
 - Organization culture and structure
 - Cultural differences, background

Team Dynamics – Good or Bad?

In Agile development, is team dynamics a good or bad thing?
Discuss

Team Dynamics – Pros and Cons

- Can be good
 - E.g., Improve overall team performance (productive conflict, different perspectives)
- Can be bad
 - Can lead to unproductive conflict can demotivate and prevent team from achieving its goals

Team Dynamics – Tuckman Team Development Model



Image: <https://www.atlassian.com/agile/teams>

Tuckman's stages of group development - https://en.wikipedia.org/wiki/Tuckman%27s_stages_of_group_development

Team Dynamics – Identification and Resolution

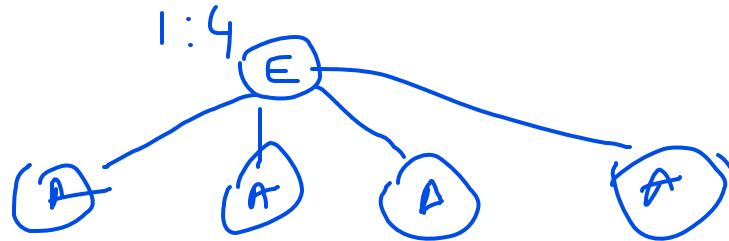
- Result from the interaction of many factors
 - E.g., Personalities, work style, roles, culture, organizational structure
- Investigate the root causes of conflict or poor team performance
 - Structured interviews or informal chats in a private and confidential
- Identify potential improvements
 - E.g., change in office layout, team development workshops (practices,
³ personality dynamics, ⁴ cultural change programs)¹ ²



Source: <https://mysoftwarequality.wordpress.com/2014/09/04/cross-dysfunctional-teams>

Agile Teams – Skills

- It's claimed that agile teams work with the best developers. However, this is not necessary the case;
- Every project needs at least one experienced and competent lead person (Critical Success Factor)
- Each experienced and competent person on the team permits the presence of 4-5 “average” or learning people
- With that skill mix, agile techniques have been shown to work many times

Skill mix
preference



E: experienced 
A: average 

Teams Culture

- Consider the following scenarios:
 - *Developer*: “It works in our environments, it’s operations responsibility to make it work in production”
 - *Tester*: “Listen, it worked in User Acceptance Testing, it must be a configuration issue, or a missing firewall hole and nothing I could have spotted during testing...”
 - *Customer*: “Hello! Nothing works here...”
- Are these statements signs of team dynamics?
- Should this kind of culture exist in agile teams?

Yes, differing views
No!

Source: <https://mysoftwarequality.wordpress.com/2014/09/04/cross-dysfunctional-teams>

Effective Teamwork

- Teamwork comprises of the right tools, the right people and the right practices
- Effective teamwork is everyone's shared responsibility
- Large software organizations, teams involve many roles across different departments (engineering, design, sales/marketing, legal)
- Use team building activities to build effective teams

Team Building

- “Various types of activities used to enhance social relations and define roles within teams, often involving collaborative tasks”
- There is an evidence how team building affect positively team effectiveness

Team Building Activities

- **Goal setting:** emphasizes the importance of clear objectives and individual and team goals
 - E.g., objectives and key results (Atlassian)
- **Interpersonal relations:** focus on teamwork skills such as giving and receiving support, communication and sharing information
- **Standup meetings**
- **Roles and responsibilities** (Atlassian)

Team Building – Roles and Responsibilities

- Define the roles and responsibilities that will make your team successful
- Clarify expectations as a team
- Helps to move a team from “storming” to “norming”, or help “performing” teams to get back on track

Roles and Responsibilities – How?

- Create a table of roles and responsibilities
 - Responsibilities from own perspective
 - Responsibilities from team member's perspective

Other's
Perspective



| Name | Role | Responsibility (own) | Responsibility (other's) |
|------|------|----------------------|---|
| CD | PO | Backlog | Backlog + communicating with client etc.. |
| . | . | . | . |
| . | . | . | . |

Roles and Responsibilities – How?

- Identify roles
 - E.g., team lead, developers, designers
 - Coarse-grained
 - Add to the role's column
- Clarify own responsibility
 - Think of top 3-5 tasks in priority order
 - Write on sticky notes
- Think of teammate's responsibilities
 - Write 1-2 responsibilities for each role from your perspective
 - Write responsibilities you may think that don't have a clear owner

Roles and Responsibilities – How?

- Refine and consolidate (optional)
 - Talk teammates with similar roles and refine responsibilities
- Discuss all roles
 - Role owner(s) describe their role and place their sticky notes in own responsibility column
 - Other teammate's role description in the other column
 - Owner to accept/decline the responsibilities by other teammates (suggest role to own it). Define primary owner for overlapping roles
 - Add “unassigned responsibilities” to
- Summarize roles and responsibilities
 - All to agree
 - Owner to document it and how to fill skill gaps

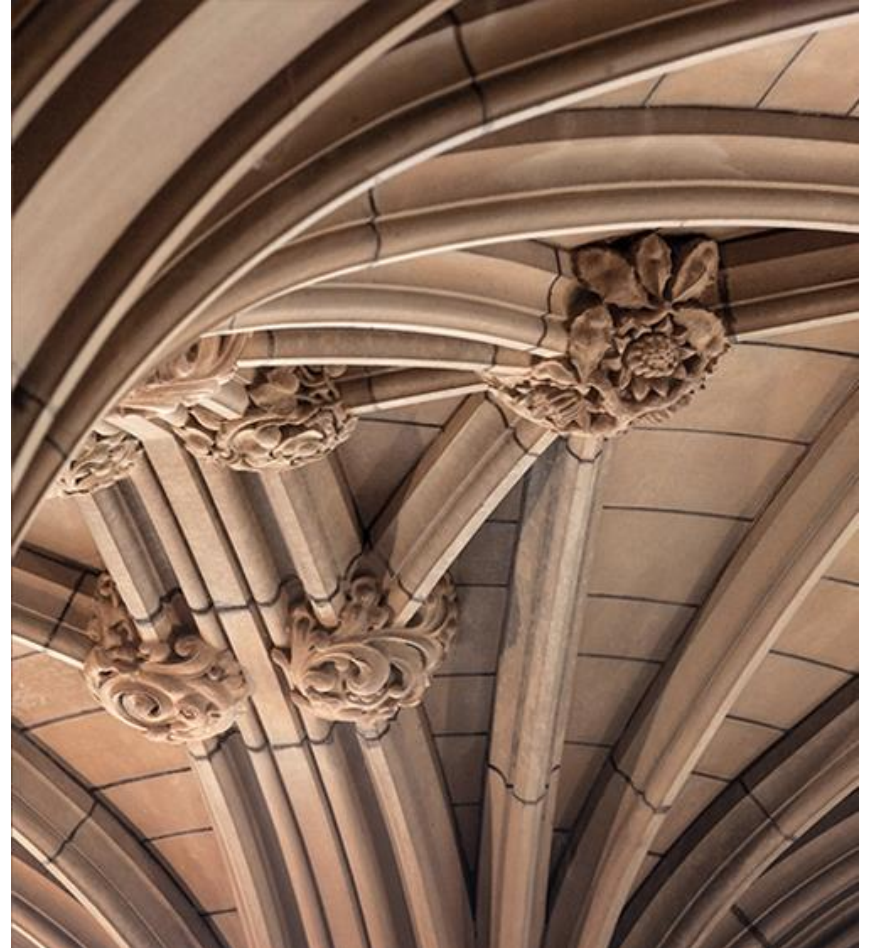
High-Performing Agile Teams

- Cross-functional; engineers, designers, architects, sales
- Mutual respect and mutual responsibility
 - Not blaming culture, and/or “throw it over the wall”
- Sound engineering practices (tools and automation)
- Value and belief of agile practices and principles
- Apply agile practices effectively as individuals and as a team
- Receive continuous training (technical and non-technical) and team monitoring/coaching

How much can you find out?

- Search for:
 - Team effectiveness
 - Self-managed teams
 - Group conflict
 - Team efficacy...
- Learn about the theory of teams!

Tools and Technologies for Teamwork



Issue Tracking Systems

- A software that manages and maintains lists of issues
- Used to create, update and resolve reported issues internally or externally
- **Bug (defect) Tracking System:** keeps track of reported software bugs in software development projects
 - Centralized overview of development requests and their states
 - May assigned a priority, status, severity and/or complexity
 - Prioritized list of pending items (Backlog)
 - Typically integrated with other tools or software management systems

↓ Trello, github issues etc...

https://en.wikipedia.org/wiki/Issue_tracking_system

https://en.wikipedia.org/wiki/Bug_tracking_system

Bug/Issue Tracking Part of other Systems

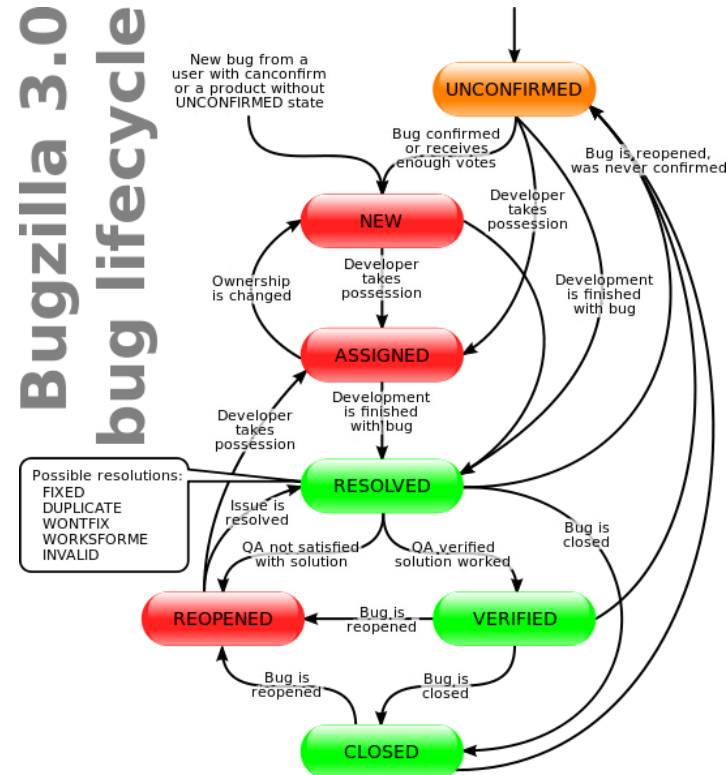
- Part of integrated project/software development management systems
- It helps integrating issue/bug tracking with other activities
- Distributed bug tracking tools are designed to be used with distributed revision control software

Bugzilla – Bug Tracking Tool

- **Open-source** web-based bug tracker and testing tool by Mozilla project
- Bug (or feature) requests can be submitted by anyone and will be assigned to a particular developer
- Various status updates for each bug
 - E.g., Bugzilla itself allows the public to file bugs – it assigns all bugs to a gatekeeper whose job is to assign responsibility and priority level


<https://en.wikipedia.org/wiki/Bugzilla>


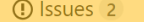





Bugzilla – Bug Lifecycle




By Nyco [GFDL (<http://www.gnu.org/copyleft/fdl.html>) or CC BY-SA 3.0 (<https://creativecommons.org/licenses/by-sa/3.0/>)], from Wikimedia Commons

Issue Tracking – GitHub

 SOFT2412-Agile-Software-Development / **Test** Private

 Code  Issues 2  Pull requests 0  Projects 1  Wiki  Insights  Settings



Login bug

Write

Preview


When a user enters a username or password with non-permitted characters such as a semicolon


Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Styling with Markdown is supported

Submit new issue

Assignees

 bsul6138

 asan0483

Labels

bug

Projects

Test repo

Milestone

No milestone

The University of Sydney

Page 45

GitHub Issue Tracking – Edit Labels

SOFT2412-Agile-Software-Development / Test Private

Watch 0 Star 0 Fork 0

Code Issues 2 Pull requests 0 Projects 1 Wiki Insights Settings

Search all labels Labels Milestones New label

8 labels Sort

| | | |
|------------------|---|-------------|
| bug | Something isn't working | Edit Delete |
| duplicate | This issue or pull request already exists | Edit Delete |
| enhancement | New feature or request | Edit Delete |
| good first issue | Good for newcomers | Edit Delete |
| help wanted | Extra attention is needed | Edit Delete |
| invalid | This doesn't seem right | Edit Delete |
| question | Further information is requested | Edit Delete |
| wontfix | This will not be worked on | Edit Delete |

SOFT2412-Agile-Software-Development/Test/labels

GitHub Issue Tracking – Project Management

Enterprise This organization Search Pull requests Issues Explore

SOFT2412 - Agile Software Development Practices

Repositories 54 People 79 Teams 20 Projects 1 Settings

Test Agile Updated 3 hours ago

Filter cards + Add cards (1 new) Fullscreen Menu

5 To do

Enter a note

Add Cancel

☒ Search filter user story
Added by bsul6138

☒ Product Search User Story
Added by bsul6138

☒ Automation
Automatically move your cards to the right place based on the status and activity of your issues and pull requests.
Added by ffar6831

☒ Welcome to GitHub Projects
We're so excited that you've decided to create a new project! Now that you're here, let's make sure you know how to get the most out of GitHub Projects.

1 In progress

☒ Test 2 - added
Added by hoso5448

2 Done

☒ test
Test#2 opened by ffar6831

☒ Complete User Sign-up
Test#3 opened by bsul6138

+ Add column

Version Control Systems – GitHub Revisit

- GitHub allows groups of people to collaborate across many projects at the same time in organizations account
- Organization's members can be:
 - **Owner:** complete administrative access to the organization
 - **Member:** default role for everyone else
- Owners can manage members' access to the organization's repos. and projects with fine-grained permission controls
- Can add collaborators from outside of the organization (consultant) to have access to one or more organization repos. without bring a member of the organization

GitHub – Organization Access Control (Revisit)

| Organization action | Owners | Members |
|---|--------|---------|
| Invite people to join the organization | X | |
| Edit and cancel invitations to join the organization | X | |
| Remove members from the organization | X | |
| Reinstate former members to the organization | X | |
| Add and remove people from all teams | X | |
| Promote organization members to <i>team maintainer</i> | X | |
| Add collaborators to all repositories | X | |
| Access the organization audit log | X | |
| Delete all teams | X | |
| Delete the organization account, including all repositories | X | |

| Organization action | Owners | Members |
|---|--------|---------|
| Create teams | X | X |
| See all organization members and teams | X | X |
| @mention any visible team | X | X |
| Can be made a <i>team maintainer</i> | X | X |
| Transfer repositories | X | |
| View a project board and add or reorganize its cards and columns | X | X |
| Create or delete a project board and edit its description | X | X |
| Automate actions for project boards | X | X |
| View and post private team discussions to all teams (see " About team discussions " for details) | X | |
| Edit and delete team discussions in all teams (for more information, see " Managing disruptive comments ") | X | |

– Examples of access permissions for organization's owners and members

<https://help.github.com/enterprise/2.13/user/articles/permission-levels-for-an-organization/>

GitHub – Creating Organization (Revisit)

The screenshot shows the GitHub 'Sign up your team' page on the domain `github.sydney.edu.au`. On the left, a user profile dropdown menu is open, showing options: 'New repository', 'New gist', and 'New organization' (which is highlighted in blue). The main heading is 'Sign up your team'. Below it, a progress bar shows three steps: 'Step 1: Completed Create personal account', 'Step 2: Create organization' (the current step, indicated by a blue icon), and 'Step 3: Add members'. The section 'Create an organization account' contains a form with the following fields:

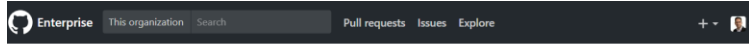
- Organization name:** A text input containing 'COMP9201' with a green checkmark on the right. Below it, a note states: 'This will be your organization name on `https://github.sydney.edu.au/COMP9201`'.
- Contact email:** A text input containing 'basem.suleiman@sydney.edu.au'.

A green 'Create organization' button is located below the form. To the right of the form, a text block explains: 'Organization accounts allow your team to plan, build, review, and ship software — all while tracking bugs and discussing ideas.'

Organizational accounts have a namespace where all their projects exist

This screenshot shows the GitHub interface for the organization 'COMP9201' on the domain `github.sydney.edu.au`. The browser's address bar shows the URL `https://github.sydney.edu.au/COMP9201`. The top navigation bar includes the GitHub logo, the word 'Enterprise', and a dropdown menu currently showing 'This organization'. Other navigation links include 'Search', 'Pull requests', 'Issues', and 'Explore'.

GitHub – Add Members to Organization (Revisit)



Add organization members

✓ Completed
Create personal account

👤 Step 2:
Create organization

👥 Step 3:
Add members

Search by username, full name or email address

hoso5448

hoso5448 Hamzah Bin Osop

Farnaz Farid

Finish

Organization members

✓ See all repositories ⓘ

✓ Create repositories

✓ Organize into teams

✓ Review code

✓ Communicate via @mentions

As an organization owner, you'll have complete access to all of the organization's repositories and have control of what members have access using fine-grained permissions.

- Note: when you create a new repo you can create them under your personal account or under any of the organizations that you're owner in

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

Repository name

COMP9201

/

WebStoreApp

Choose another owner

bsul6138

✓ COMP9201

INFO3220-Object-Oriented-Design

PROFESSIONAL-Software-Engineering

SOFT2201-Software-Constr-and-Design-1

SOFT2412-Agile-Software-Development

Need inspiration? How about

choose who can commit.

repository.

GitHub Organization – Manage Repos. (Revisit)

The screenshot shows the GitHub Organization page for 'COMP9201'. The top navigation bar includes 'Enterprise', 'This organization', 'Search', 'Pull requests', 'Issues', and 'Explore'. The main content area has a header with the organization name and a 'Create a new repository' button. Below this, there are tabs for 'Repositories', 'People', 'Teams', 'Projects', and 'Settings'. The 'Repositories' tab is active, showing a list of repositories: 'Front-end' (Private), 'Designs' (Private), 'Back-end' (Private), and 'WebStoreApp' (Private). Each repository entry includes a brief description and the last update time. On the right side, there is a 'People' section with a list of team members: 'bsul6138' (Basem Fathi Suleiman) and 'ffar6831' (Farnaz Farid).

The screenshot shows the 'Teams' page within the GitHub Organization. The top navigation bar is the same as the previous screenshot. The main content area has a header with the organization name and a 'Create new team' button. Below this, there are tabs for 'Options', 'Collaborators & teams', 'Hooks', 'Integrations & services', 'Deploy keys', and 'Custom tabs'. The 'Teams' tab is active, showing a list of teams: 'FrontEndDeve' (Front-end development team, 2 members, Read permission), 'BackEndDeve' (Back-end Development Team, 2 members, Admin permission), and 'Designers' (Web Application Designers, 1 member, Read permission). Each team entry includes a brief description and the last update time. At the bottom, there is a button to 'Add a team: Select team'.

GitHub Organization – Manage People (Revisit)

COMP9201

Repositories 0 People 3 Teams 0 Projects 0 Settings

Find a member...

Members Outside collaborators Add member

| Select all | 2FA | Role | 0 teams |
|----------------------------------|-------|----------------|---------|
| Basem Fathi Suleiman bsul6138 | 2FA X | Private Owner | 0 teams |
| Farnaz Farid ffar6831 | 2FA X | Private Member | 0 teams |
| Hamzah Bin Osop hoso5448 | 2FA X | Private Member | 0 teams |

- Manage
- Change role...
- Convert to outside collaborator
- Remove from organization

Enterprise This organization Search Pull requests Issues Explore

COMP9201

Repositories 4 People 3 Teams 3 Projects 0 Settings

ffar6831 has access to 3 repositories

Find a repository they have access to...

| Repository | Access Level | Manage access |
|----------------------|--------------------------|---------------|
| COMP9201/WebStoreApp | Read on this repository | Manage access |
| COMP9201/Designs | Write on this repository | Manage access |
| COMP9201/Front-end | Admin on this repository | Manage access |

Role: Member

3 repositories

1 team




Membership private

Two-factor security disabled

Convert to outside collaborator

Remove from organization

GitHub Organization – Manage Teams (Revisit)

| | | |
|--|---|----------|
| Find a team... | Import teams | New team |
| Select all | | |
| Visibility Members | | |
| BackEndDeve Back-end Development Team |  2 members | 0 teams |
| Designers Web Application Designers |  1 member | 0 teams |
| FrontEndDeve Front-end development team |  2 members | 0 teams |

You may have 3 repos; Designs, Front-end and Back-end. You want FrontEndDeve to work on the Front-end and Designs repos, Designers team to work on Designs repo and BackEndDeve to work on Back-end repo

COMP9201 / BackEndDeve

Discussions

Members 2

Teams 0

Repositories 2

Find a repository...

Add repository

| | | |
|------------------------|---------|-------|
| Select all | | |
| COMP9201/Back-end | Private | Admin |
| updated 21 minutes ago | | |
| COMP9201/WebStoreApp | Private | Admin |
| updated an hour ago | | |

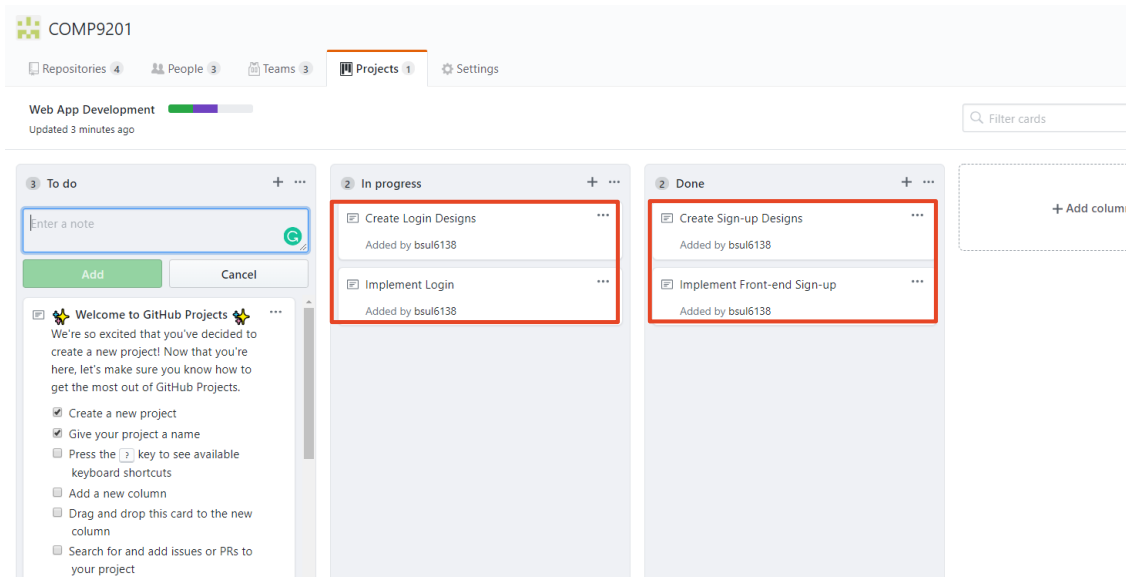
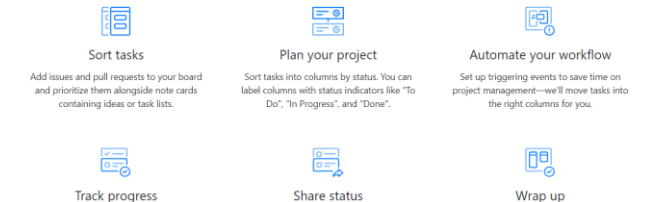
GitHub Organization – Manage Projects (Revisit)



Organize your issues with project boards

Did you know you can manage projects in the same place you keep your code? Set up a project board on GitHub to streamline and automate your workflow.

[Learn More](#) [Create a project](#)



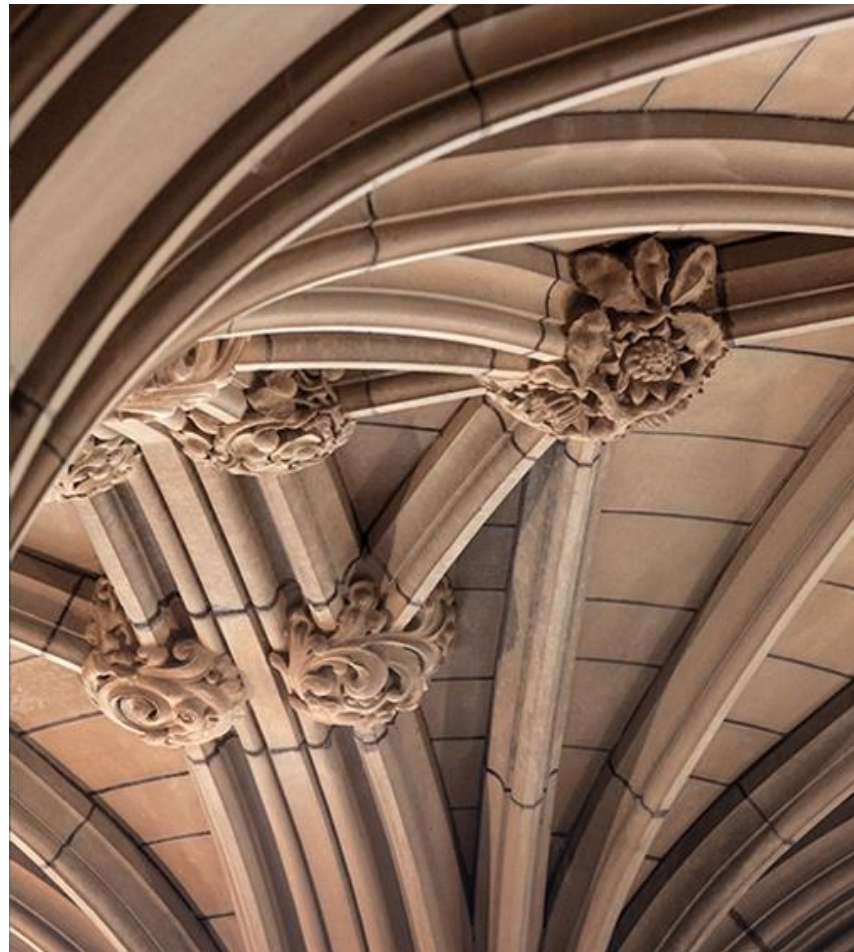
GitHub Organization – Audit Log (Revisit)

- Audit log records all events that have happened at the organization level, who did them and where in the world they were done

The screenshot shows the GitHub Organization Audit Log for the organization COMP9201. The top navigation bar includes links for Repositories (4), People (3), Teams (3), Projects (1), and Settings. The left sidebar lists various organization settings, with 'Audit log' selected. The main content area displays the 'Audit log' with a search bar and a list of recent events. The events listed are:

- bsul6138 – project.create**: Created project [Web App Development](#) in COMP9201. Location: [Australia](#). Time: an hour ago.
- bsul6138 – team.add_repository**: Gave COMP9201/frontenddeve access to COMP9201/Designs. Location: [Australia](#). Time: an hour ago.
- bsul6138 – team.add_repository**: Gave COMP9201/frontenddeve access to COMP9201/WebStoreApp. Location: [Australia](#). Time: an hour ago.
- bsul6138 – team.add_repository**: Gave COMP9201/designers access to COMP9201/WebStoreApp. Location: [Australia](#). Time: an hour ago.
- bsul6138 – team.add_repository**: Gave COMP9201/backenddeve access to COMP9201/WebStoreApp. Location: [Australia](#). Time: an hour ago.

Jenkins



Jenkins – CI / CD



- “Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.”
- *Jenkins pipeline* is a suite of plugins which supports implementing and integrating *continuous delivery pipelines* into Jenkins
 - A *continuous delivery pipeline* is an automated expression of your process for getting software from version control right through to end users/customers
 - Typically written in *Jenkinsfile* which is checked in a project’s source code repository

Jenkins – Integration with GitHub (1)

The screenshot shows the Jenkins web interface in a browser. The address bar indicates the URL is 192.168.9.1/view/all/newJob. The Jenkins logo and name are visible in the top left, and a search bar and user information (soft2412) are in the top right. The main content area is titled 'Enter an item name' and contains a text input field with the value 'jenkins-github'. Below the input field is a list of project types, each with an icon and a description:

- Freestyle project**: This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Pipeline**: Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**: Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- GitHub Organization**: Scans a GitHub organization (or user account) for all repositories matching some defined markers.
- Multibranch Pipeline**: Creates a set of Pipeline projects according to detected branches in one SCM repository.

At the bottom, there is a section titled 'If you want to create a new item from other existing, you can use this option:' with a 'Copy from' button and a text input field for 'Type to autocomplete'.

Jenkins – Integration with GitHub (2)

Source Code Management

☐ None
☒ Git

Repositories

Repository URL

https://github.sydney.edu.au/SOFT2412-Agile-Software-Development/git-local5.git

Credentials

hoso5448/*****

Add

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/master

Add Branch

Repository browser

(Auto)

Additional Behaviours

Add

☐ Subversion

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☐ Build after other projects are built

☐ Build periodically

☐ GitHub Branches

☐ GitHub Pull Requests

☒ GitHub hook trigger for GITScm polling

☐ Poll SCM

Jenkins – Integration with GitHub (3)

- Webhooks to set up GitHub applications to subscribe to certain events on GitHub
- Events is triggered, HTTP POST payload will be sent to the webhook's configured URL
- Webhooks can be used to update an external issue tracker, trigger CI builds, update a backup mirror

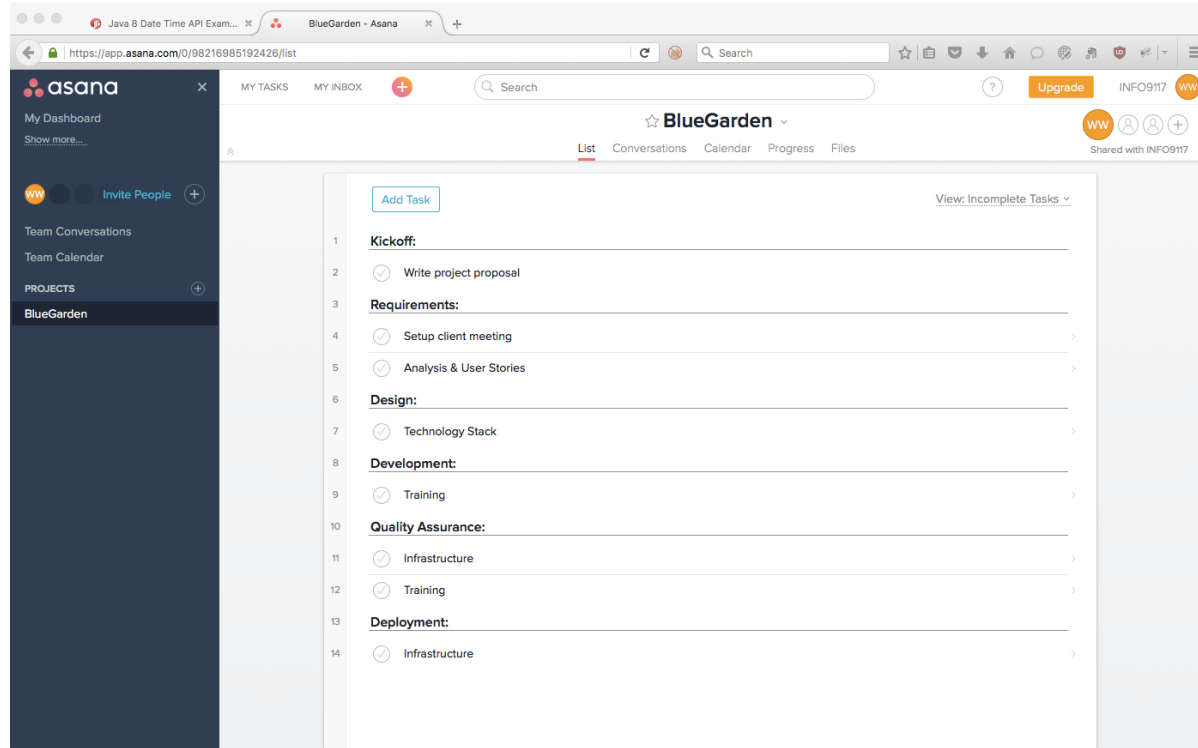
The screenshot shows the GitHub 'Manage webhook' interface. On the left, a sidebar contains links: Options, Collaborators & teams, Branches, Hooks (selected), Integrations & services, Deploy keys, and Custom tabs. The main content area is titled 'Webhooks / Manage webhook' and includes a description of webhooks. It contains several input fields: 'Payload URL' with the value 'http://pc-226-2.staff.sydney.edu.au/github-webhook/', 'Content type' set to 'application/x-www-form-urlencoded', and an empty 'Secret' field. Below these is a section 'Which events would you like to trigger this webhook?' with three radio buttons: 'Just the push event.', 'Send me everything.', and 'Let me select individual events.' (which is selected). Under the selected option, there are two columns of checkboxes for various events: Commit comments, Branch or tag deletion, Deployment statuses, Wiki, Issues, Branch or tag creation, Deployments, Forks, Issue comments, and Labels. Each checkbox is accompanied by a brief description of the event.

<https://developer.github.com/webhooks/>

Teamwork Collaboration Tools

- What tools would you use to help support your team?
- Examples:
 - Dropbox
 - Google Docs
 - Skype
 - Trello
 - Slack
 - Basecamp
 - Asana
 - ... other

Other Tools – Asana – Project Management



References

- Andrew Stellman, Margaret C. L. Greene 2014. Learning Agile: Understanding Scrum, XP, Lean and Kanban (1st Edition). O'Reilly, CA, USA.
- S. P. Myers, 2013. [<https://www.teamtechnology.co.uk/team/dynamics/definition/>]
- Cross Dysfunctional Teams.
[<https://mysoftwarequality.wordpress.com/2014/09/04/cross-dysfunctional-teams>]
- Atlassian, Agile Teams. [<https://www.atlassian.com/agile/teams>]
- Tuckman's stages of group development.
https://en.wikipedia.org/wiki/Tuckman%27s_stages_of_group_development
- Atlassian, Teamwork. [<https://www.atlassian.com/teamwork>]
- Team building. [https://en.wikipedia.org/wiki/Team_building]
- Issue tracking system. [https://en.wikipedia.org/wiki/Issue_tracking_system]
- Bug tracking system. [https://en.wikipedia.org/wiki/Bug_tracking_system]

References

- Further Readings:
 - Hackman J. R., *“Leading Teams: Setting the stage for great performances”*, Harvard Business Press 2002
 - Hackman J. R., Katz N. *“Group behavior and performance”*. In Fiske ST, Gilbert DT, Lindzey G *Handbook of social psychology* (5th ed.) New York: Wiley; 2010. pp. 1208-1251. DOI: 10.1002/9780470561119.socpsy002032

Tutorial: Continuous Integration with Jenkins / Docker

Team Dynamics – Team Building Activities

Lecture: Agile Methods, Scrum

