Agile Software Development Practices

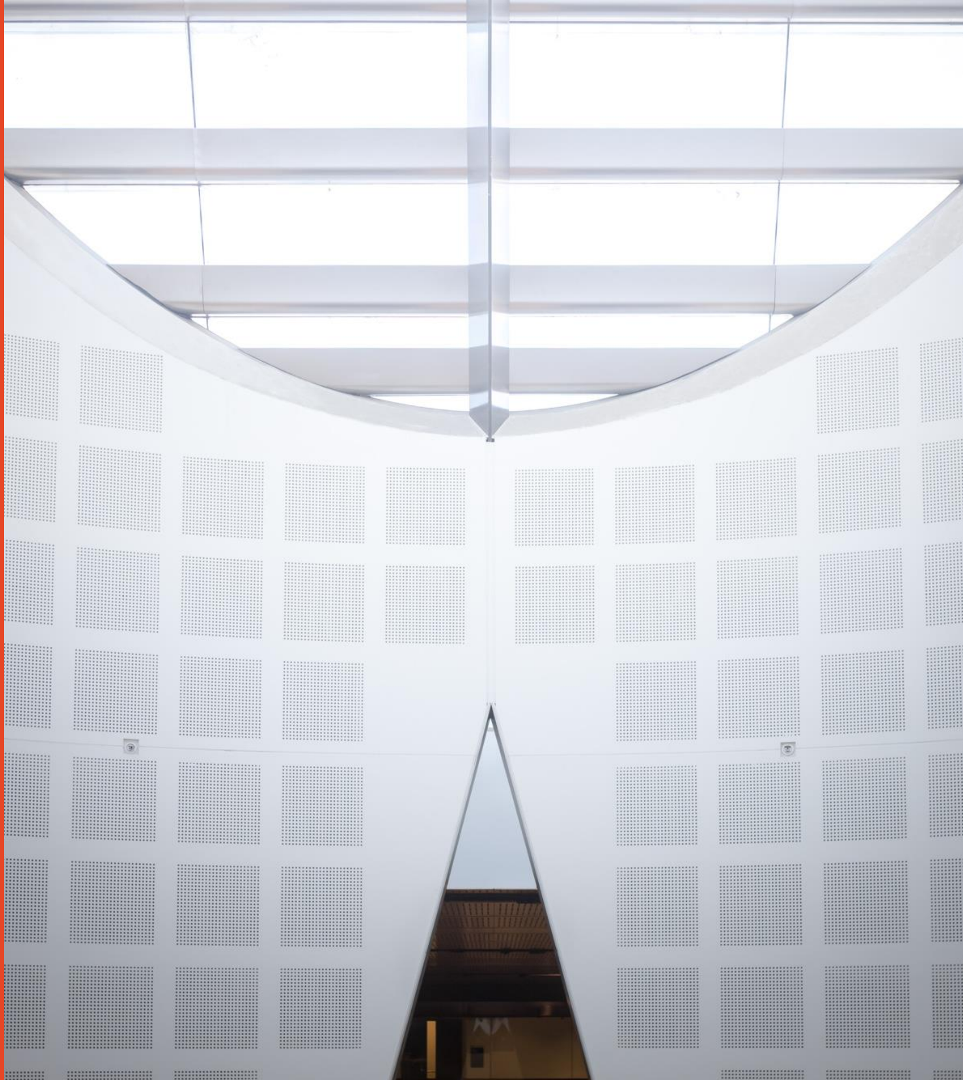
SOF2412 / COMP9412

Team Dynamics; Tools and Technologies for Teamwork

Dr. Basem Suleiman

School of Information Technologies

THE UNIVERSITY OF SYDNEY

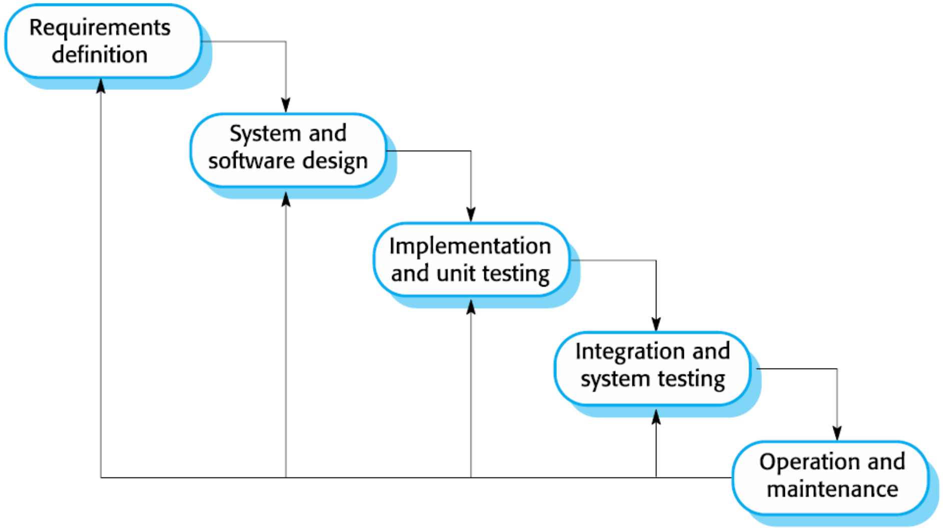


Copyright warning

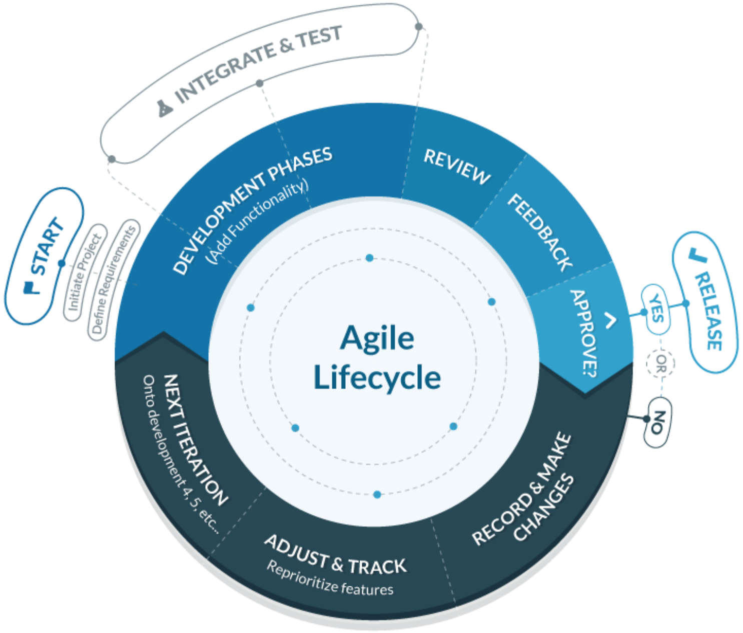
Agenda

- Teams in Agile Development
- Team Dynamics
- Tools and Technologies for Teamwork

Software Development Models – Teams


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Waterfall model
plan-driven development

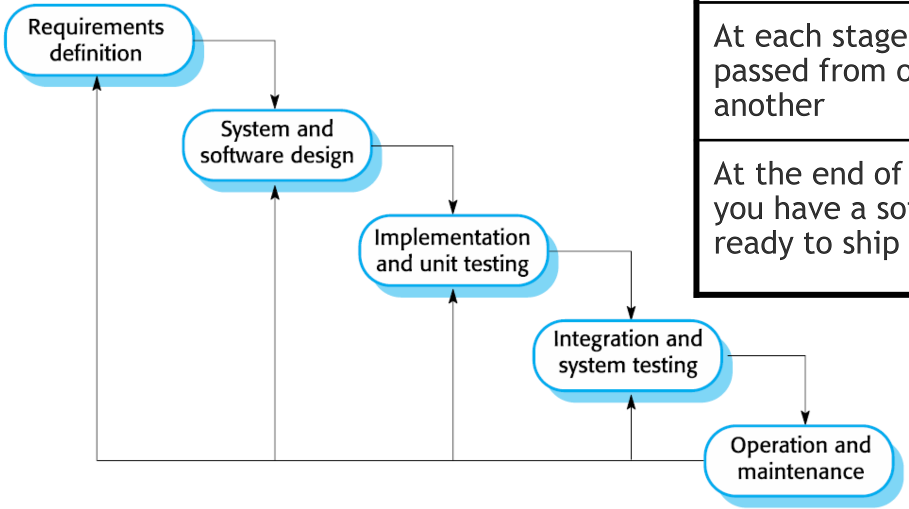


Agile model
Incremental & iterative development

Ian Sommerville. 2016. Software Engineering (10th ed. Global Edition). Pearson

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

Waterfall Model – Teams


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

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

Ian Sommerville. 2016. Software Engineering (10th ed. Global Edition). Pearson

Groups 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 / clear division of labor
    - Project management tracks progress and reallocates “resources”
    - Clear authority lines, so disagreements can be resolved
  - Problems?
    - Single points of failure
    - Inflexibility
    - Lack of feedback

- An 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

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

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 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 – People

- Business **people** and **developers** must **work together daily** throughout the project
- 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**
- The best architectures, requirements, and designs emerge from **self-organizing teams**
- At regular intervals, the **team reflects** on how to become more **effective,** then **tunes and adjusts its behavior** accordingly
- 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; developers thinks about development work, project managers focuses on managing resources, and they throw the code over the wall to a business user who thinks business
  - Individual working separately with no communication – teams are divided and collaboration is killed

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, project managers 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

- The agile elephant is made up of many practices, the whole thin is much bigger than the sum of the parts

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

Affinity Estimating
Relative sizing
FDD
Ideal time
Customer-valued prioritization
Continuous Improvement
Planning Poker
ROI
WIP limits
Test-driven development
Co-location
Wireframe
TDD
Negotiation
Information radiator
Individuals and interactions
Working Software
Scrum
Agile tooling
Self Organizing teams
MMF s
Continuous integration
Story maps
Frequent verification and validation
Timeboxing
Osmotic communication
Task boards
Test-first development
Iteration
Refactoring
Cumulative Flow Diagrams
Adaptive leadership
User stories
Relative ranking
Personas
Minimally Marketable Feature
Declaration of interdependence
Definition of done
Spike
Story points
Retrospectives
Theme
Agile Manifesto
Iterative Development
Agile modeling
Chartering
Servant leadership
Product roadmap
Relative prioritization
Cycle time
Pair Programming
Kanban boards
Wideband Delphi
Epic
Process tailoring
Emotional intelligence
Escaped defects
Velocity
Responding to change
XP
Risk-adjusted backlog
Backlog
Burn down Charts
Risk Burndown graphs
Value stream mapping
Iteration planning
DSDM Atern
Release planning
Burn up Charts
Crystal Clear
Conflict resolution
Team space
NPV
Compliance
Risk-based Spike
Progressive elaboration
Collaboration
Daily stand-ups
IRR

Team Dynamics

- What is team dynamics?
- Is team dynamics a good or bad thing?
- What about group 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
  - Nature of the team's work
  - Organization culture and structure
  - Cultural differences, background

S. P. Myers, 2013 https://www.teamtechnology.co.uk/team/dynamics/definition/

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

S. P. Myers, 2013 https://www.teamtechnology.co.uk/team/dynamics/definition/

Group vs Team (Dynamics)

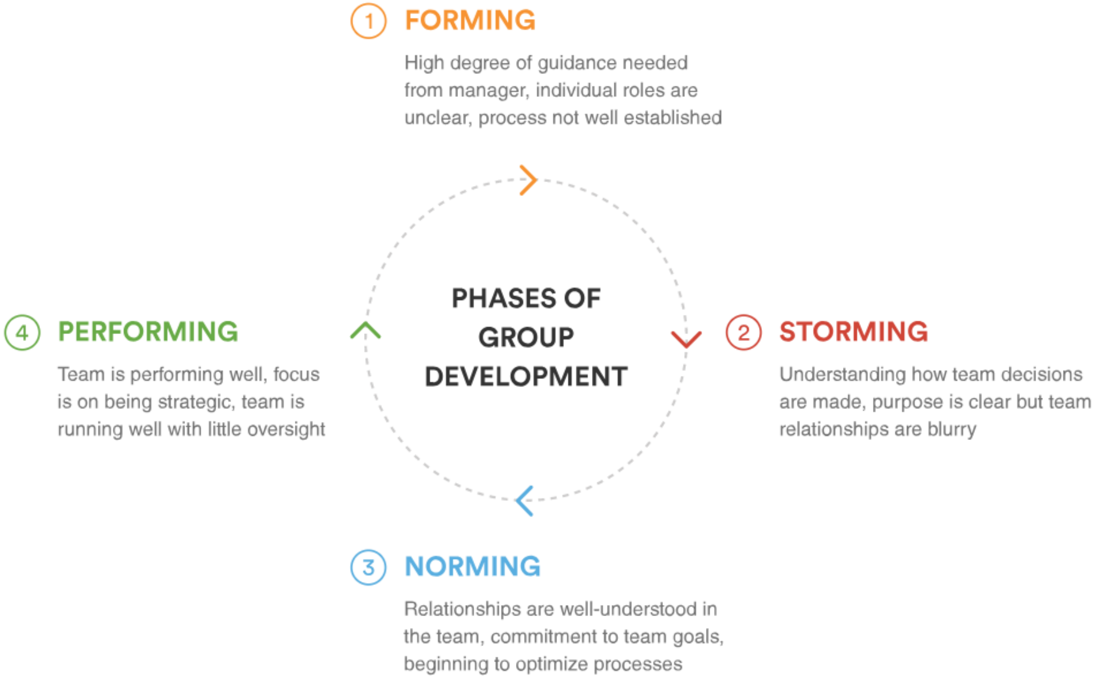
- “Groups are social community consisting of two or more people who have something in common”

- A team is special instance of a group – the commonality is the shared goal
  - Team members are dependent on each other for success (sport teams, development teams)

- The essential difference is in the terms “group” vs “team”, team dynamics and group dynamics are very similar

S. P. Myers, 2013 https://www.teamtechnology.co.uk/team/dynamics/definition/

Team Dynamics – Tuckman Team Development Model



PHASES OF GROUP DEVELOPMENT

1 FORMING

High degree of guidance needed from manager, individual roles are unclear, process not well established

2 STORMING

Understanding how team decisions are made, purpose is clear but team relationships are blurry

3 NORMING

Relationships are well-understood in the team, commitment to team goals, beginning to optimize processes

4 PERFORMING

Team is performing well, focus is on being strategic, team is running well with little oversight

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 environment
  - Individuals can share their views openly
- Based on the investigation, improvements can be identified
  - E.g., change in office layout, team development workshops (practices, personality dynamics, cultural change programs , tools to better facilitate communication/interaction

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

Team Building

- “A collective term for various types of activities used to enhance social relations and define roles within teams, often involving collaborative tasks”

- **Goal settings:** emphasizes the importance of clear objectives and individual and team goals

- **Interpersonal relations:** focus on teamwork skills such as giving and receiving support, communication and sharing information

- There is an evidence how team building affect positively team effectiveness

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

High-Performing Agile Teams

- Cross-functional; different roles in a team (engineers, designers, architects, sales)
- Built on mutual respect and mutual responsibility
  - Not blaming culture, and/or “throw it over the wall”
- Are built on sound engineering practices (tools and automation)
- Value and belief in the significance 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

Team Success – Mistaken Beliefs

- Teams that work together harmoniously perform better than those with lots of conflict

- Team dynamics are largely caused by the leader's style (authoritarian versus democratic)

- Larger teams perform better than small ones

Agile only works with the best developers

- 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

Mutual Respect

- *Developer*: “Testers are failed programmers, they shouldn’t be called engineers”

- *Tester*: “Developers are only able to produce bugs, the world would be better with more testers and less developers”

- *Business Analysts*: “I don‘t know why I bother talking to testers and developers, they are total idiots”

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

Not My Responsibility

- *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…”

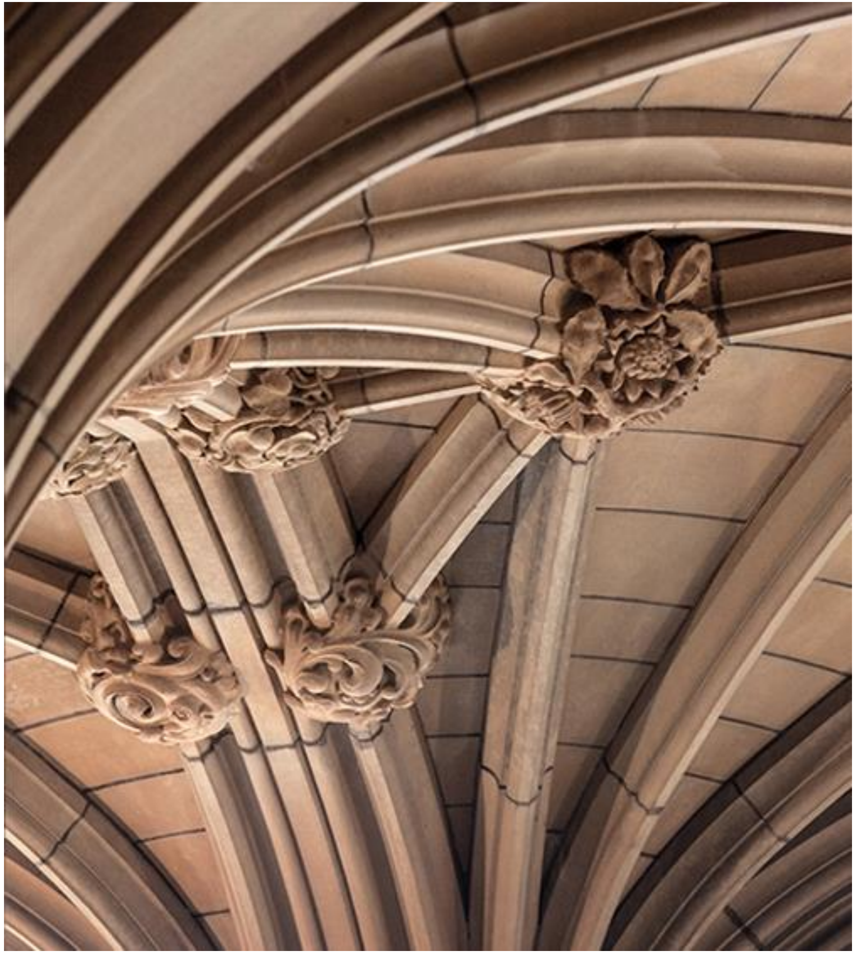
Source: https://mysoftwarequality.wordpress.com/2014/09/04/cross-dysfunctional-teams

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 (employees or departments) or externally (e.g., customer support call center)

- **Bug (defect) Tracking System:** keeps track of reported software bugs in software development projects
  - Provide centralized overview of development requests (bugs and improvements) and their states
  - May assigned a priority, status, severity and/or complexity
  - Prioritized list of pending items (Backlog) helps to define the product roadmap or next release
  - Typically integrated with other tools or software management systems

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; e.g., development, source control management

- Distributed bug tracking tools are designed to be used with distributed revision control software
  - Allow bug reports to be conveniently read, added to the database or updated while a developer is offline

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

Bugzilla – Bug Tracking Tool

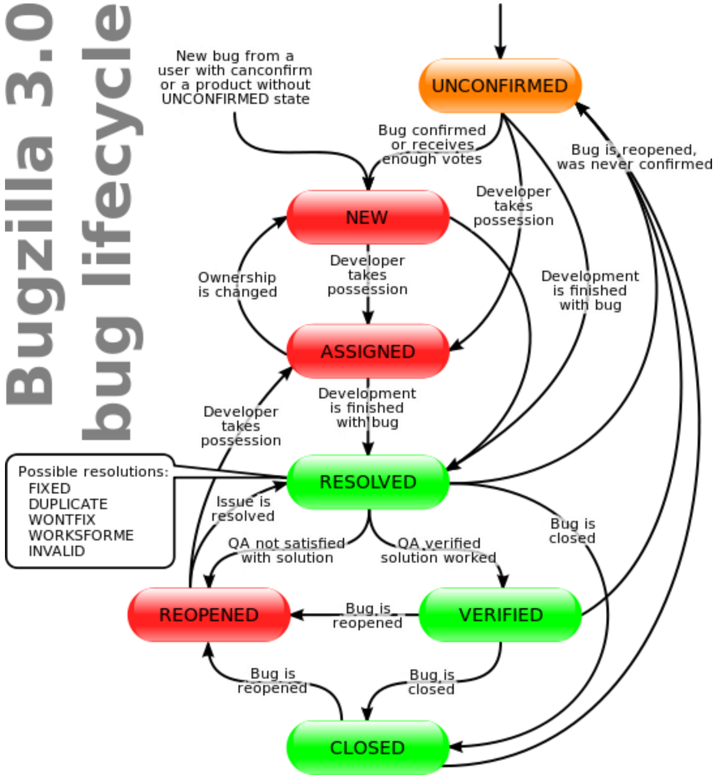
- Open-source web-based bug tracker and testing tool developed by Mozilla project
  - For free and open-source software and proprietary software

- Bug (or feature) requests can be submitted by anyone and will be assigned to a particular developer

- Various status updates for each bug are allowed
  - 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

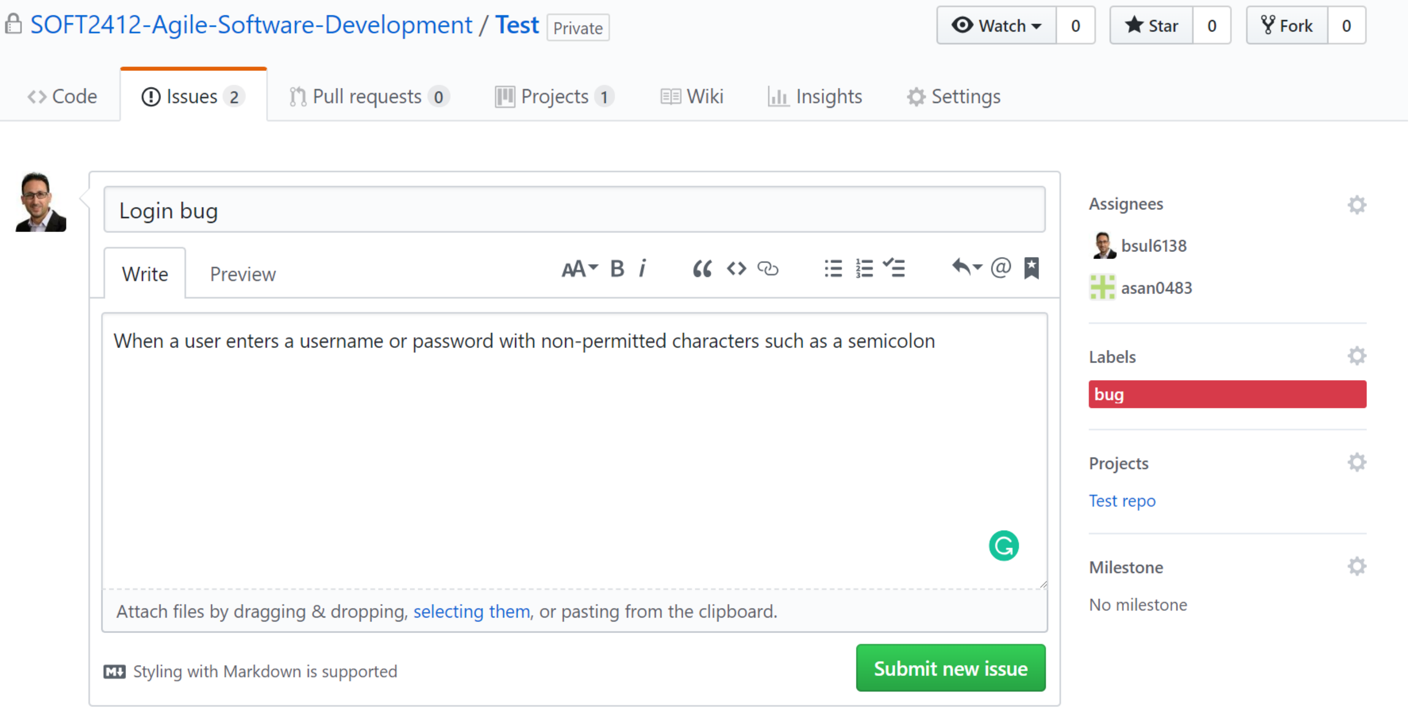
https://en.wikipedia.org/wiki/Bugzilla

Bugzilla – Bug Lifecycle


Bugzilla 3.0
bug lifecycle
New bug from a user with canconfirm or a product without UNCONFIRMED state
UNCONFIRMED
Bug confirmed or receives enough votes
Bug is reopened, was never confirmed
Developer takes possession
NEW
Ownership is changed
Developer takes possession
Development is finished with bug
ASSIGNED
Development is finished with bug
Developer takes possession
Possible resolutions:
FIXED
DUPLICATE
WONTFIX
WORKSFORME
INVALID
RESOLVED
Issue is resolved
QA not satisfied with solution
QA verified solution worked
Bug is closed
REOPENED
Bug is reopened
VERIFIED
Bug is reopened
Bug is closed
CLOSED

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



Login bug

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.

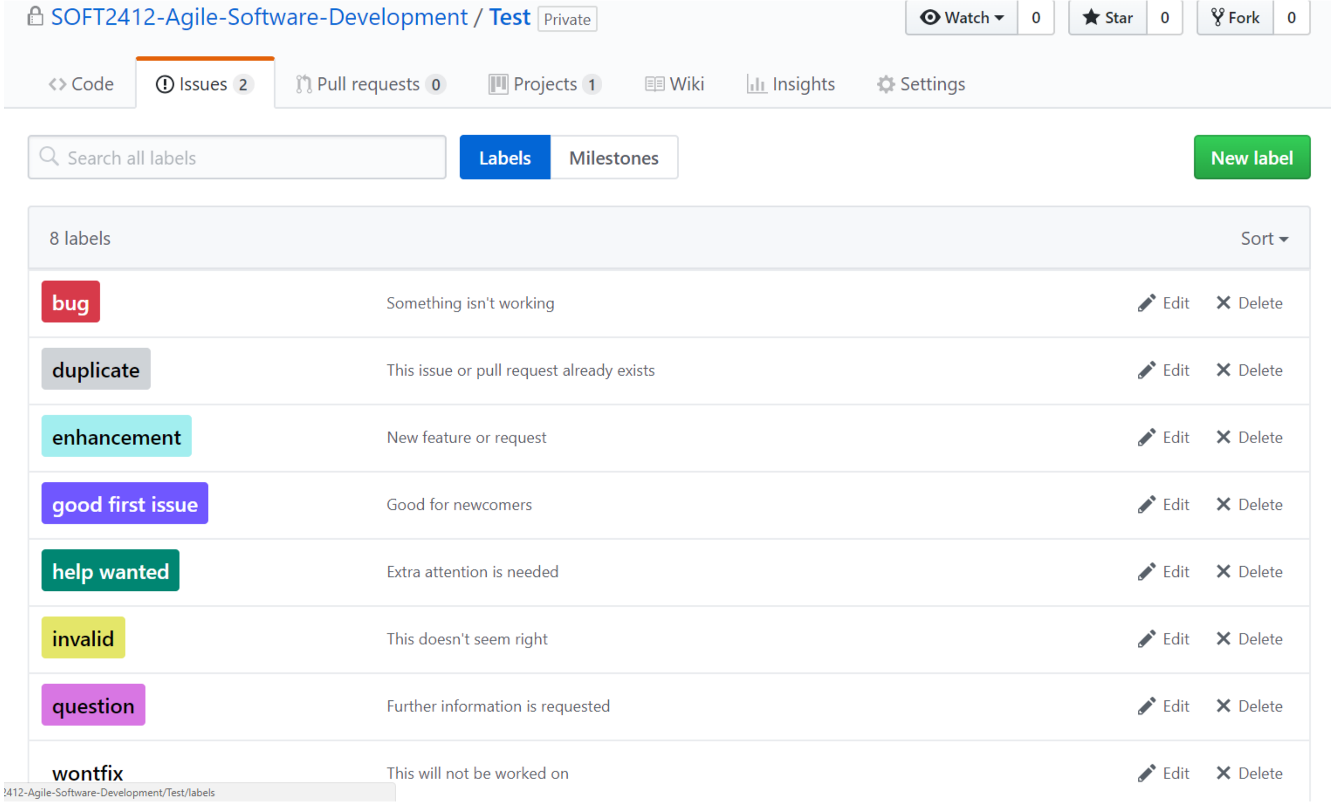
Assignees: bsul6138, asan0483

Labels: bug

Projects: Test repo

Milestone: No milestone

GitHub Issue Tracking – Edit Labels

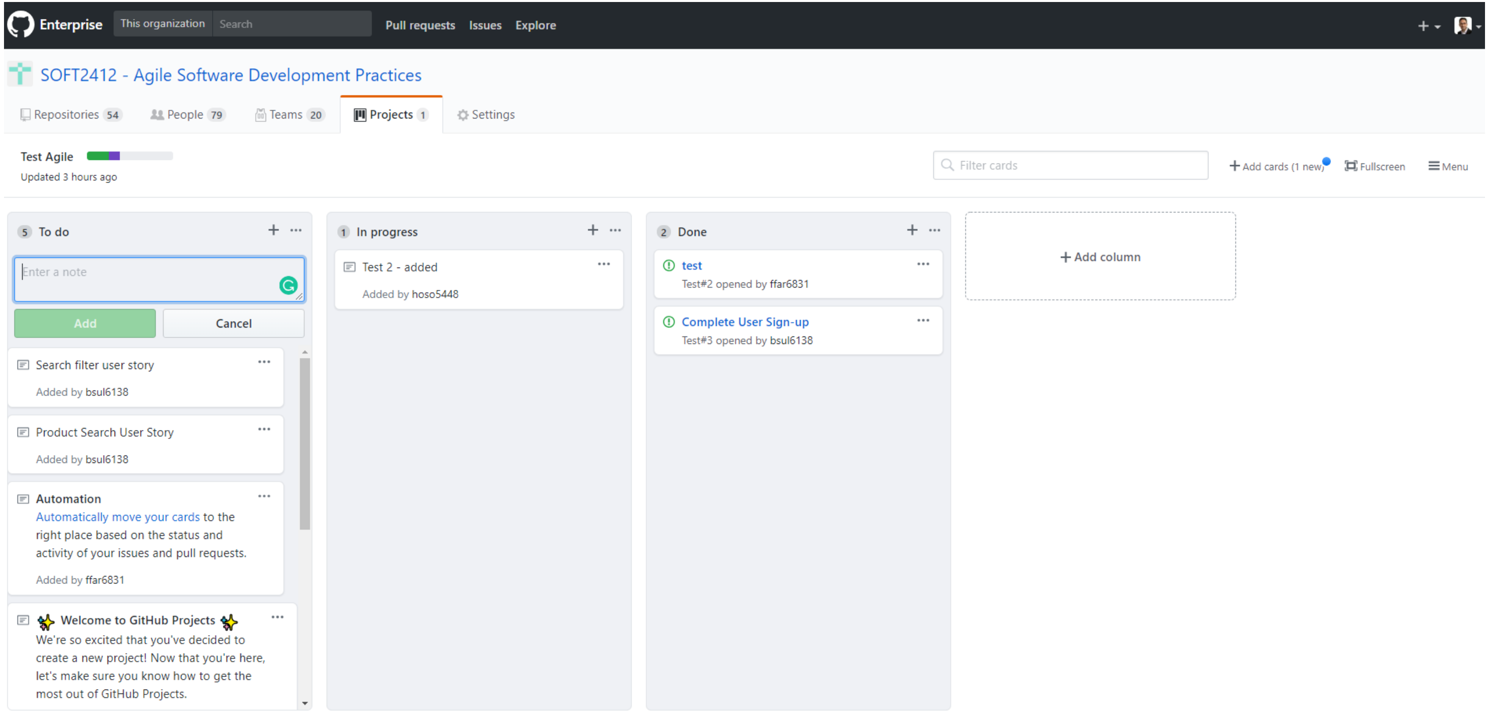


8 labels

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

2412-Agile-Software-Development/Test/labels

GitHub Issue Tracking – Project Management



SOFT2412 - Agile Software Development Practices

Test Agile

Updated 3 hours ago

5 To do

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

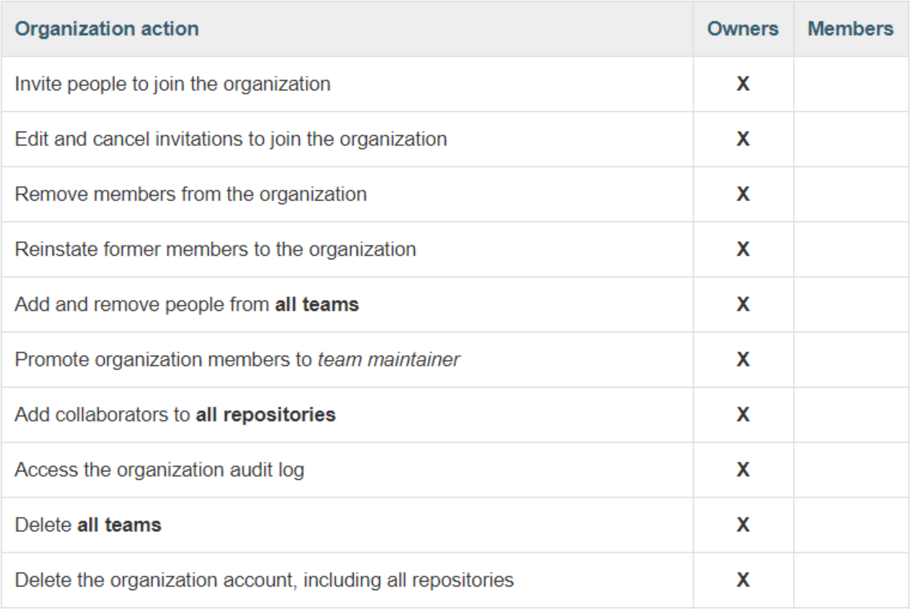
Complete User Sign-up
Test#3 opened by bsul6138

Version Control Systems – GitHub

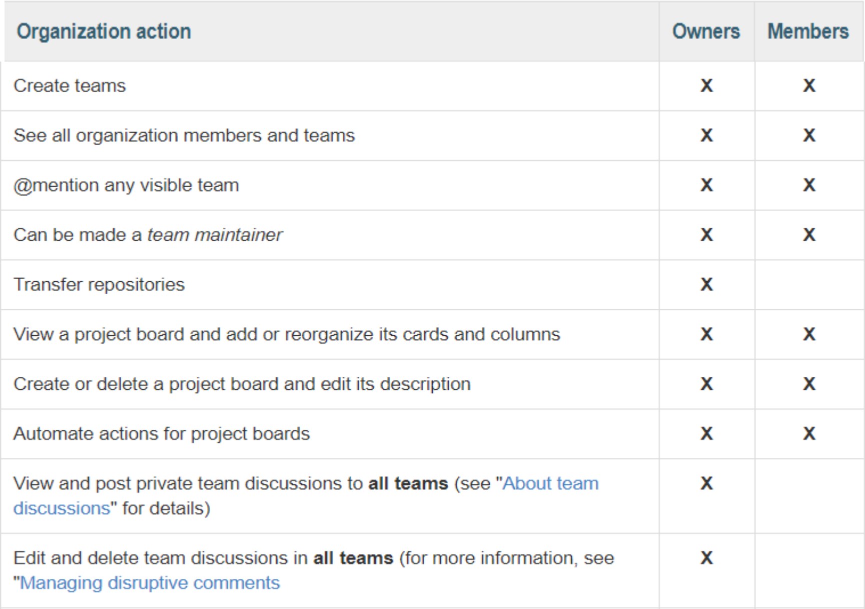
- GitHub allows groups of people to collaborate across many projects at the same time in organizations account via ***organization*** account
  - Group of people with shared ownership of projects
- Organization’s members can have owners or member roles:
  - **Owner**: have complete administrative access to the organization (often a few people in the organization should be assigned as owner roles
  - **Member**: default role for everyone else
- Owners can manage members’ access to the organization’s repos. and projects with fine-grained permission controls
  - Create your own organization
  - Understand and carefully manage members access to your organization

- 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



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 *team maintainer*	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	

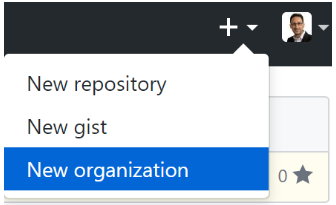


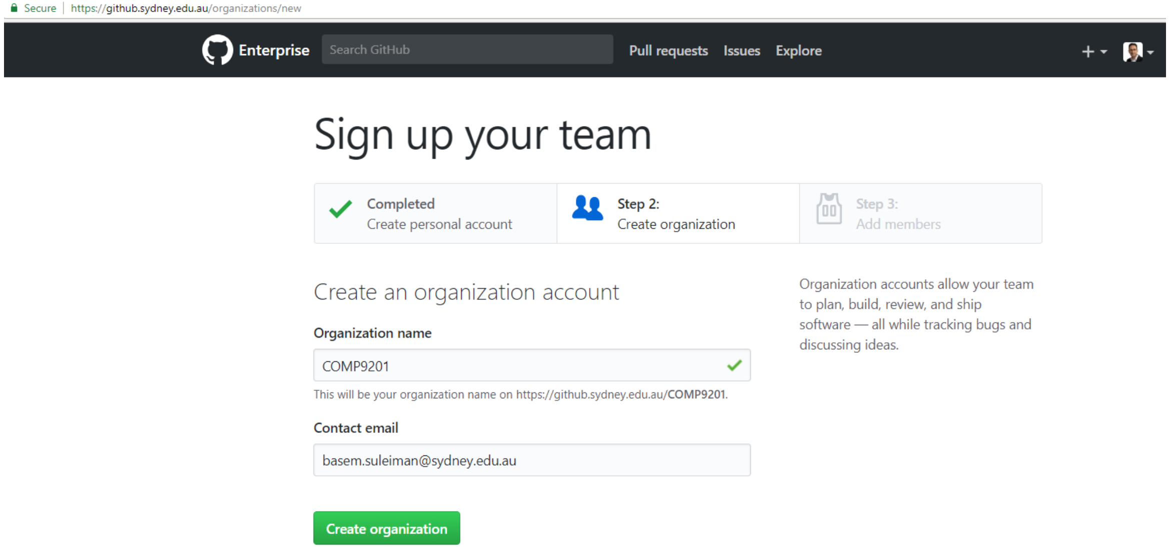
Organization action	Owners	Members
Create teams	X	X
See all organization members and teams	X	X
@mention any visible team	X	X
Can be made a *team maintainer*	X	X
Transfer repositories	X	
View a project board and add or reorganize its cards and columns	X	X
Create or delete a project board and edit its 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





Sign up your team

✓ Completed
Create personal account

Step 2:
Create organization

Step 3:
Add members

Create an organization account

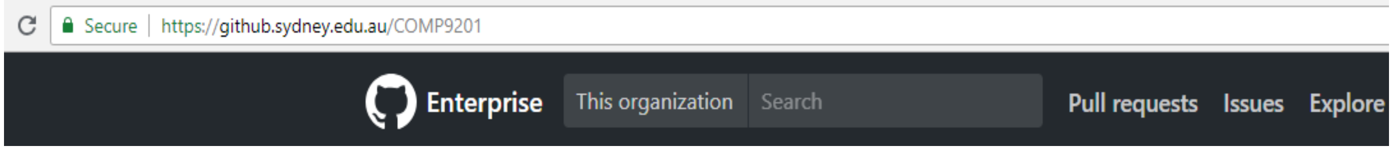
Organization name: COMP9201 ✓

This will be your organization name on https://github.sydney.edu.au/**COMP9201**.

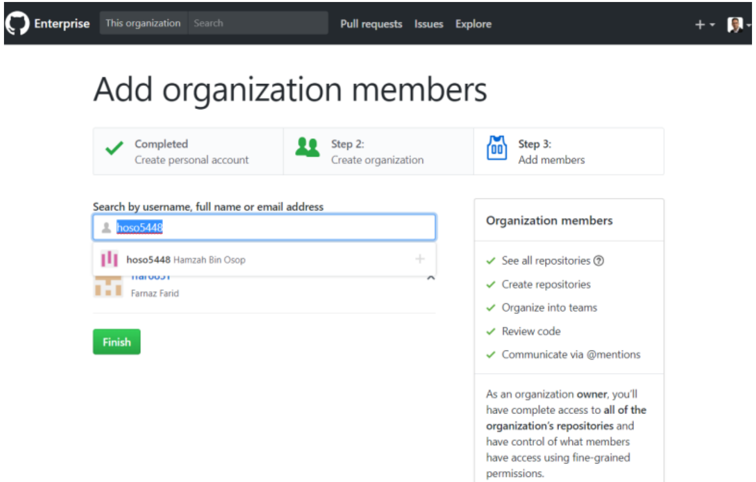
Contact email: basem.suleiman@sydney.edu.au

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



GitHub – Add Members to Organization



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

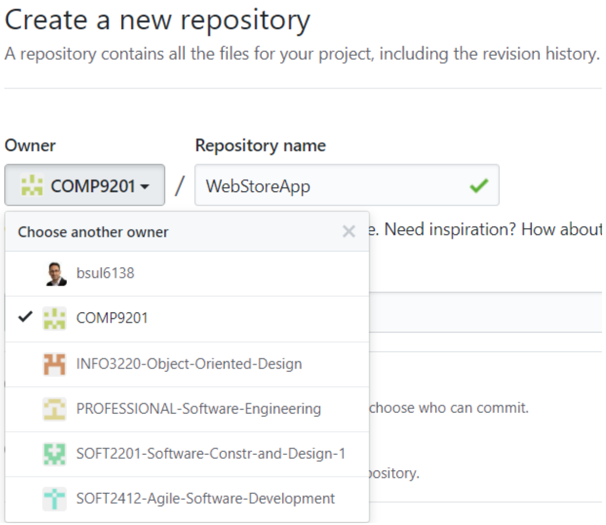
Farnaz Farid

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: COMP9201 / Repository name: WebStoreApp ✓

e. Need inspiration? How about

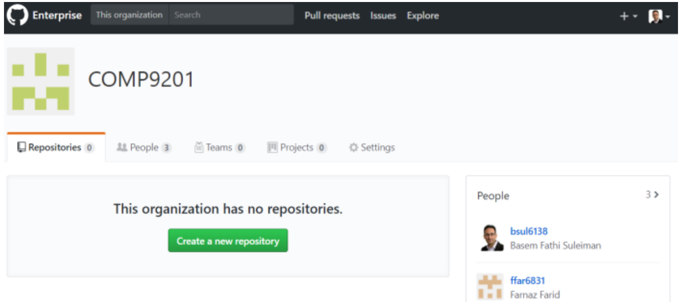
Choose another owner

- bsul6138
- ✓ COMP9201
- INFO3220-Object-Oriented-Design
- PROFESSIONAL-Software-Engineering
- SOFT2201-Software-Constr-and-Design-1
- SOFT2412-Agile-Software-Development

choose who can commit.

ository.

GitHub Organization – Manage Repos



COMP9201

Repositories 0 | People 3 | Teams 0 | Projects 0 | Settings

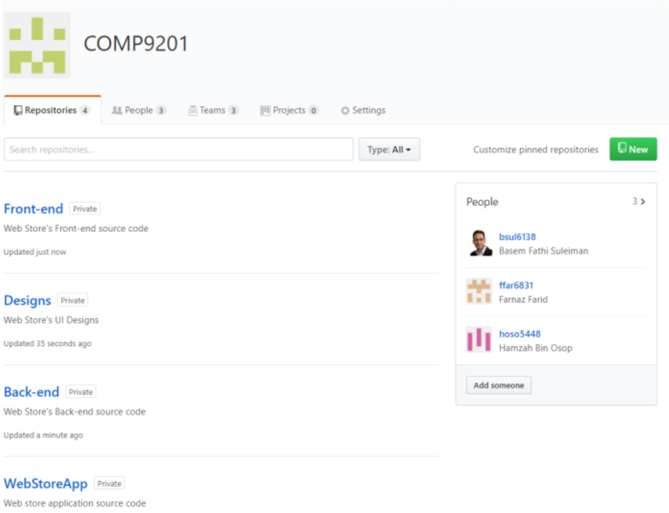
This organization has no repositories.

Create a new repository

People 3 >

bsul6138
Basem Fathi Suleiman

ffar6831
Farnaz Farid


COMP9201

Repositories 4 | People 3 | Teams 3 | Projects 0 | Settings

Front-end Private
Web Store's Front-end source code
Updated just now

Designs Private
Web Store's UI Designs
Updated 35 seconds ago

Back-end Private
Web Store's Back-end source code
Updated a minute ago

WebStoreApp Private
Web store application source code

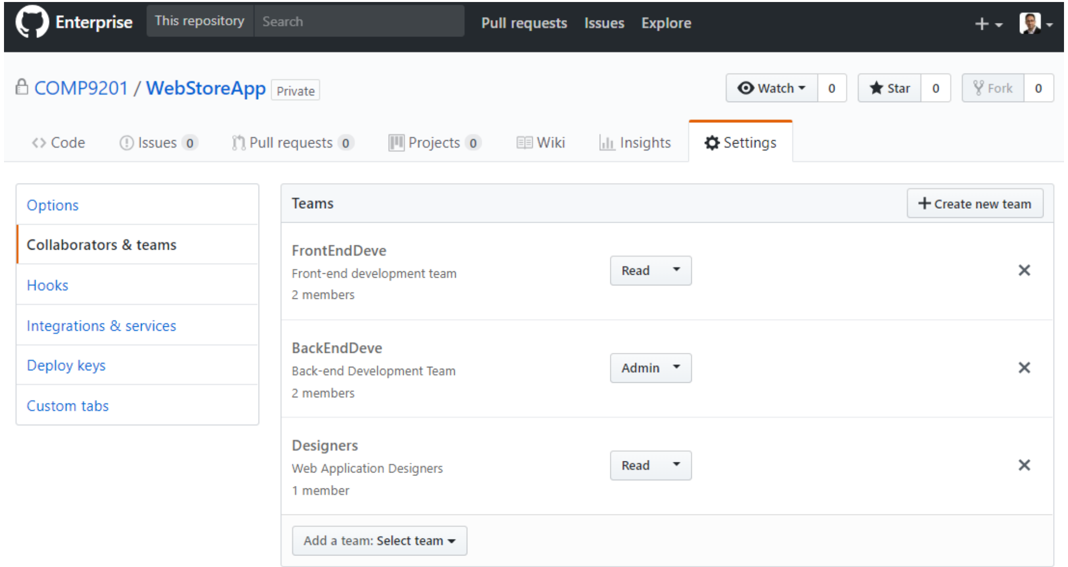
People 3 >

bsul6138
Basem Fathi Suleiman

ffar6831
Farnaz Farid

hoso5448
Hamzah Bin Osop

Add someone



COMP9201 / WebStoreApp Private

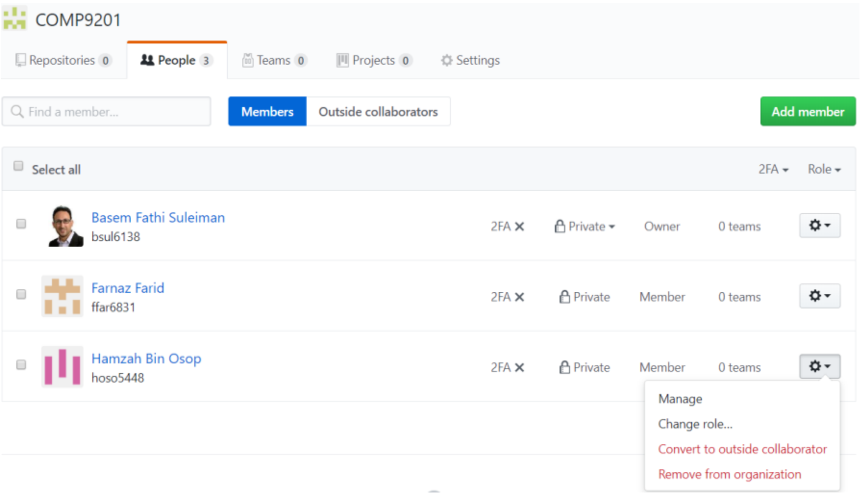
Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

Teams

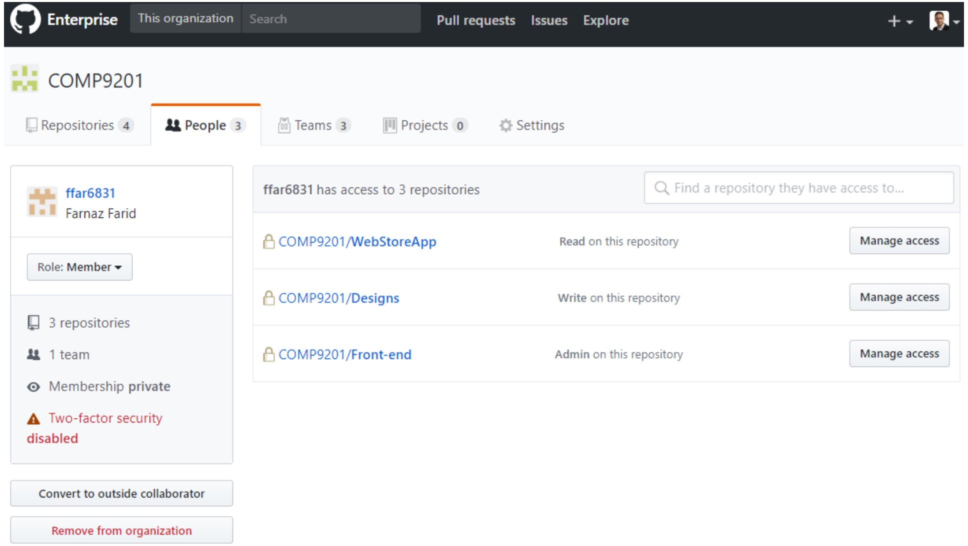
FrontEndDeve Front-end development team 2 members	Read	✕
BackEndDeve Back-end Development Team 2 members	Admin	✕
Designers Web Application Designers 1 member	Read	✕

Add a team: Select team

GitHub Organization – Manage People



Basem Fathi Suleiman bsul6138	2FA ✗	Private	Owner	0 teams
Farnaz Farid ffar6831	2FA ✗	Private	Member	0 teams
Hamzah Bin Osop hoso5448	2FA ✗	Private	Member	0 teams



ffar6831
Farnaz Farid

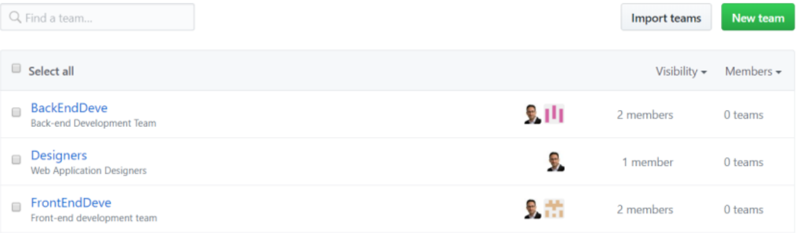
Role: Member

3 repositories
1 team
Membership private
Two-factor security disabled

ffar6831 has access to 3 repositories

COMP9201/WebStoreApp	Read on this repository
COMP9201/Designs	Write on this repository
COMP9201/Front-end	Admin on this repository

GitHub Organization – Manage Teams

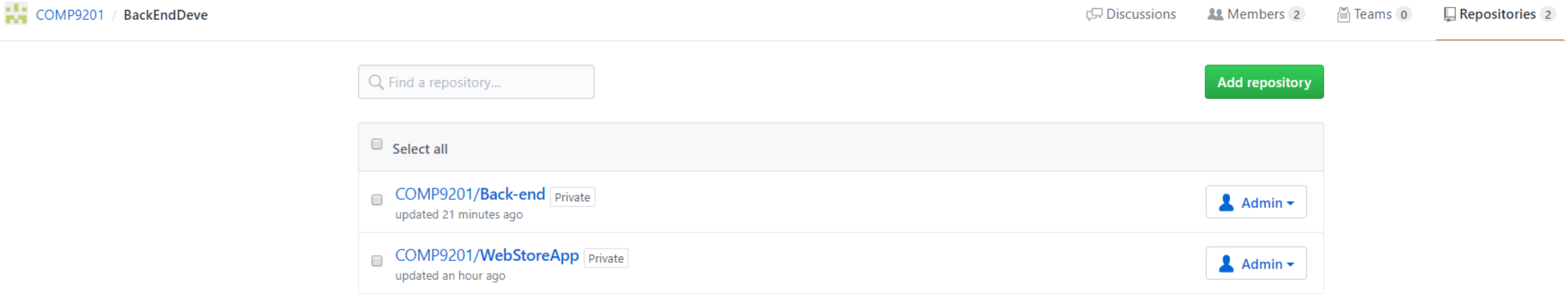


Find a team...

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...

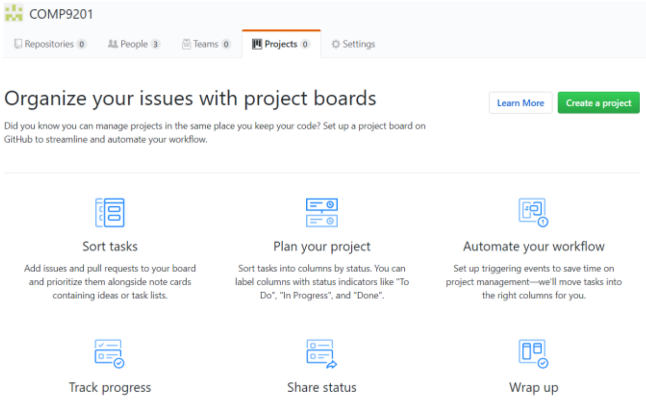
Add repository

□ Select all

□ COMP9201/Back-end Private
updated 21 minutes ago — Admin ▾

□ COMP9201/WebStoreApp Private
updated an hour ago — Admin ▾

GitHub Organization – Manage Projects



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.

Sort tasks

Add issues and pull requests to your board and prioritize them alongside note cards containing ideas or task lists.

Plan your project

Sort tasks into columns by status. You can label columns with status indicators like "To Do", "In Progress", and "Done".

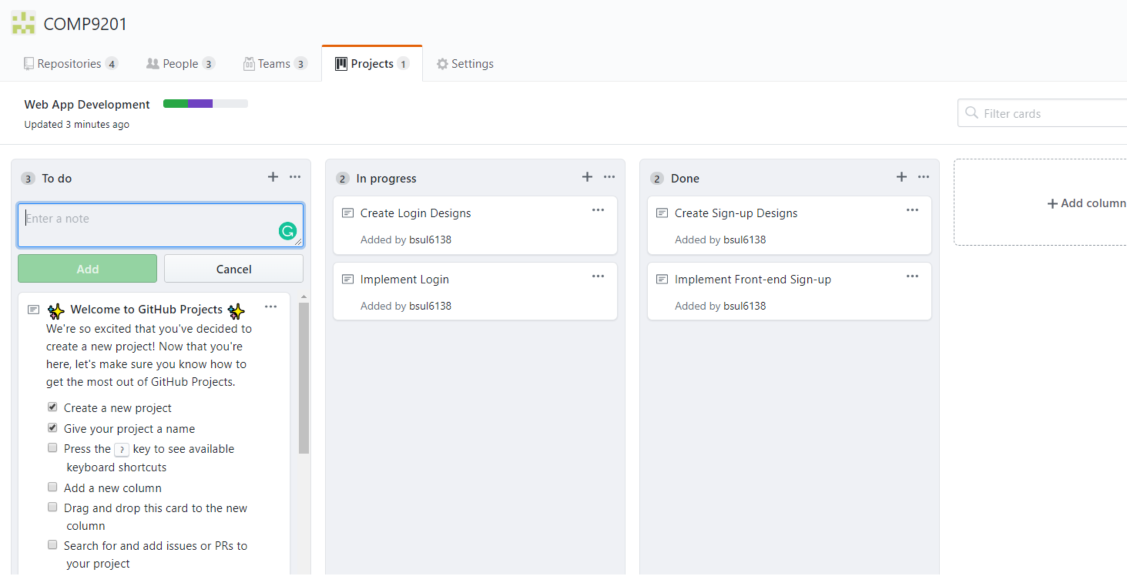
Automate your workflow

Set up triggering events to save time on project management—we'll move tasks into the right columns for you.

Track progress

Share status

Wrap up



Web App Development

Updated 3 minutes ago

3 To do

Enter a note

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.

- ☑ Create a new project
- ☑ Give your project a name
- ☐ Press the ? key to see available keyboard shortcuts
- ☐ Add a new column
- ☐ Drag and drop this card to the new column
- ☐ Search for and add issues or PRs to your project

2 In progress

Create Login Designs
Added by bsul6138

Implement Login
Added by bsul6138

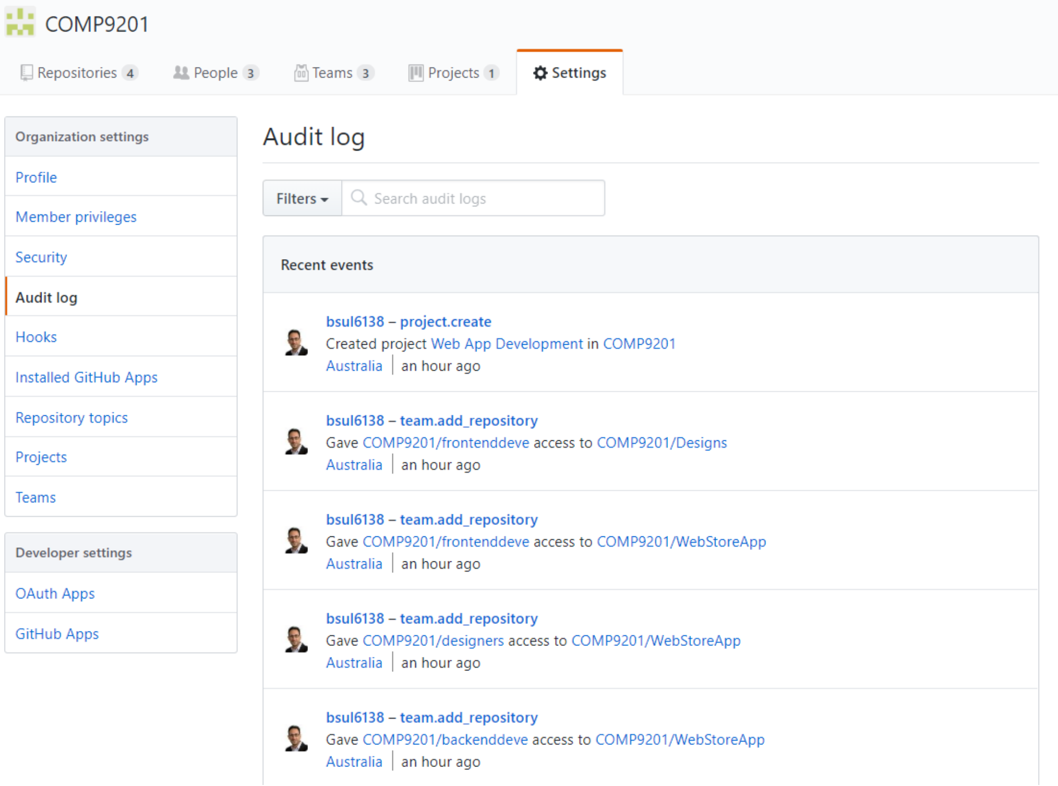
2 Done

Create Sign-up Designs
Added by bsul6138

Implement Front-end Sign-up
Added by bsul6138

GitHub Organization – Audit Log

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



Audit log

Recent events

bsul6138 – project.create
Created project Web App Development in COMP9201
Australia | an hour ago

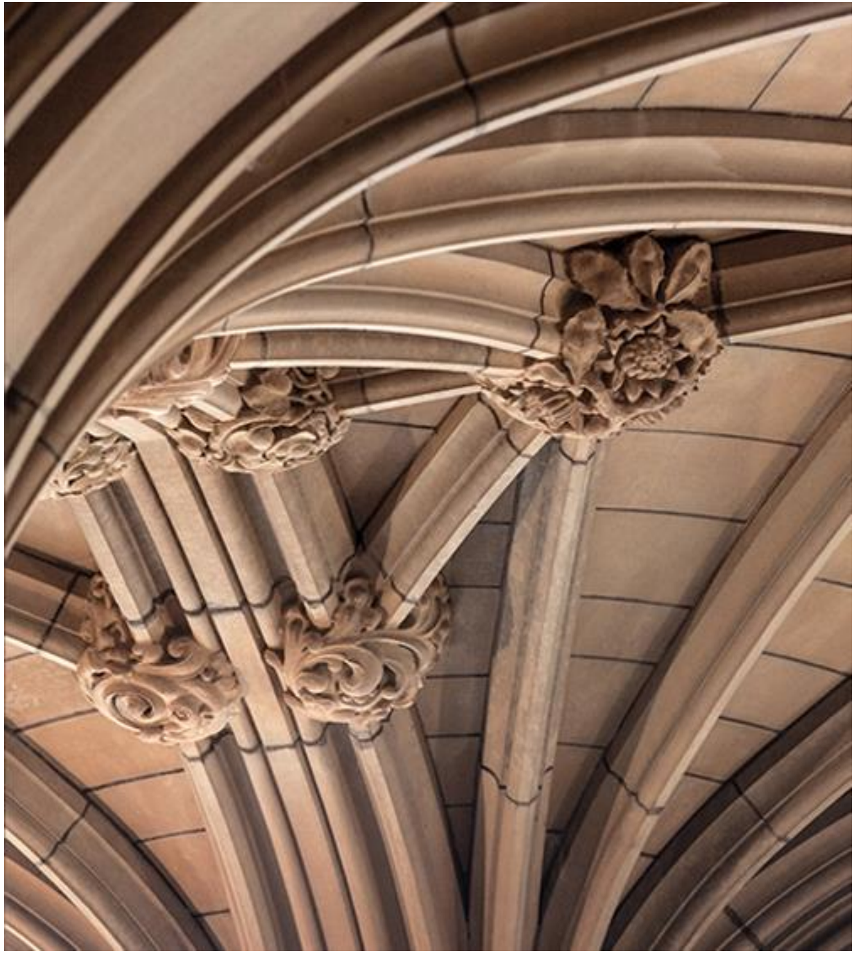
bsul6138 – team.add_repository
Gave COMP9201/frontenddeve access to COMP9201/Designs
Australia | an hour ago

bsul6138 – team.add_repository
Gave COMP9201/frontenddeve access to COMP9201/WebStoreApp
Australia | an hour ago

bsul6138 – team.add_repository
Gave COMP9201/designers access to COMP9201/WebStoreApp
Australia | an hour ago

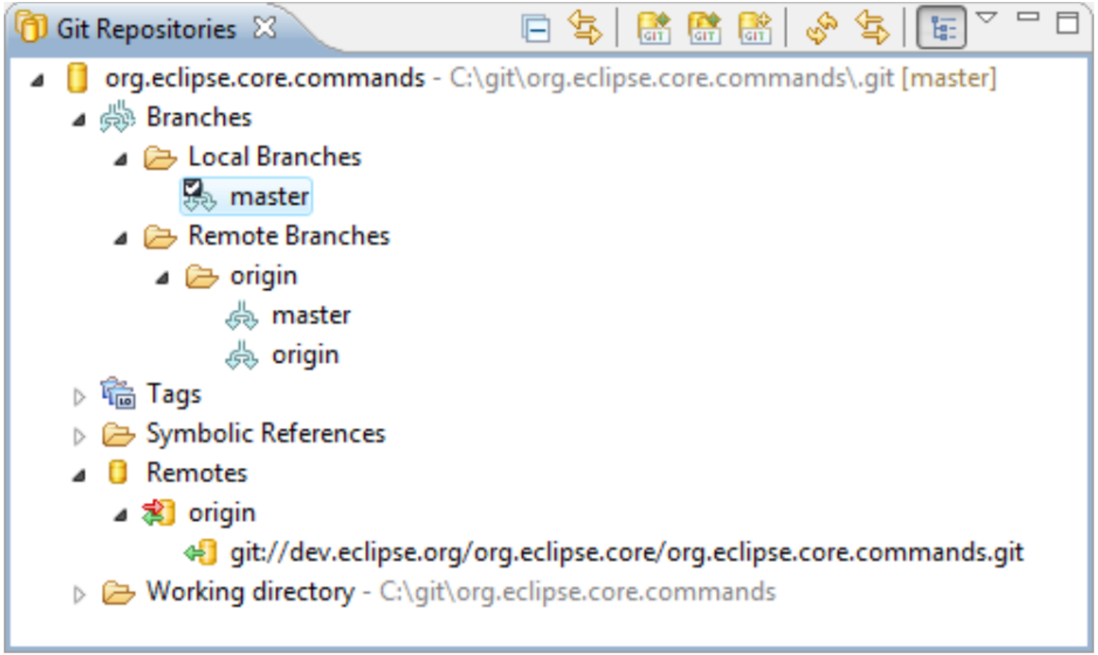
bsul6138 – team.add_repository
Gave COMP9201/backenddeve access to COMP9201/WebStoreApp
Australia | an hour ago

Git in Development Environments



THE UNIVERSITY OF SYDNEY

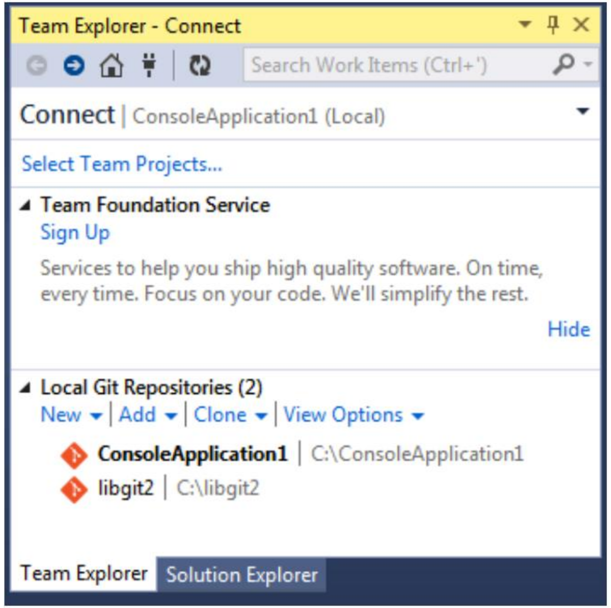
Eclipse Plugin – Egit



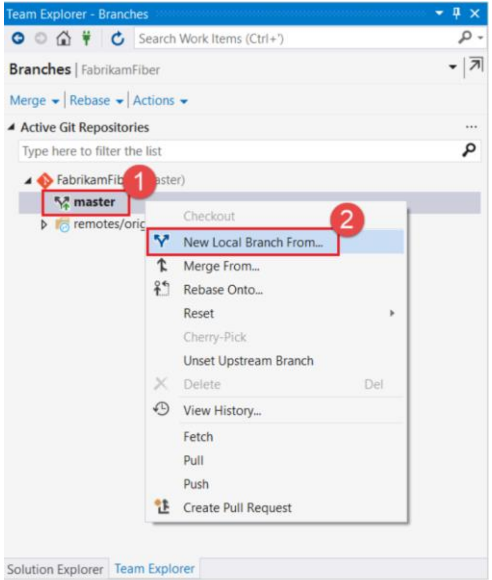
- org.eclipse.core.commands - C:\git\org.eclipse.core.commands\.git [master]
  - Branches
    - Local Branches
      - master
    - Remote Branches
      - origin
        - master
        - origin
  - Tags
  - Symbolic References
  - Remotes
    - origin
      - git://dev.eclipse.org/org.eclipse.core/org.eclipse.core.commands.git
  - Working directory - C:\git\org.eclipse.core.commands

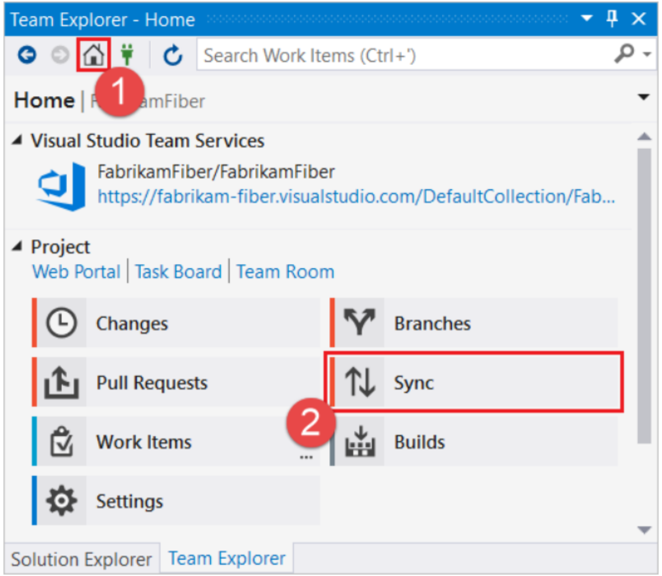
https://www.eclipse.org/egit/

Git in Visual Studio


Team Explorer - Connect
Search Work Items (Ctrl+')
Connect | ConsoleApplication1 (Local)
Select Team Projects...
Team Foundation Service
Sign Up
Hide
Local Git Repositories (2)
New | Add | Clone | View Options
ConsoleApplication1 | C:\ConsoleApplication1
libgit2 | C:\libgit2
Team Explorer
Solution Explorer

Services to help you ship high quality software. On time, every time. Focus on your code. We'll simplify the rest.

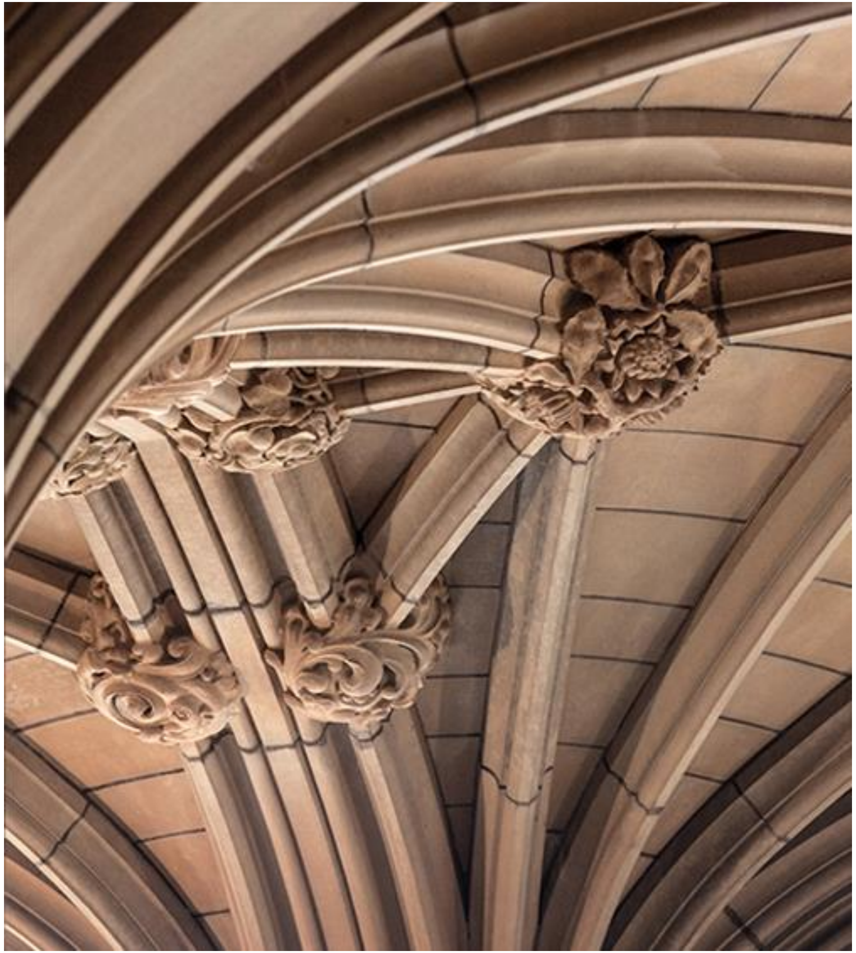



Team Explorer - Home
Search Work Items (Ctrl+')
1
Home | F amFiber
Visual Studio Team Services
FabrikamFiber/FabrikamFiber
https://fabrikam-fiber.visualstudio.com/DefaultCollection/Fab...
Project
Web Portal | Task Board | Team Room
Changes
Branches
Pull Requests
Sync
2
Work Items
Builds
Settings
Solution Explorer
Team Explorer

https://docs.microsoft.com/en-us/vsts/repos/git/gitquickstart?view=vsts&tabs=visual-studio

Jenkins

Continuous Integration / Continues Delivery/ Deployment

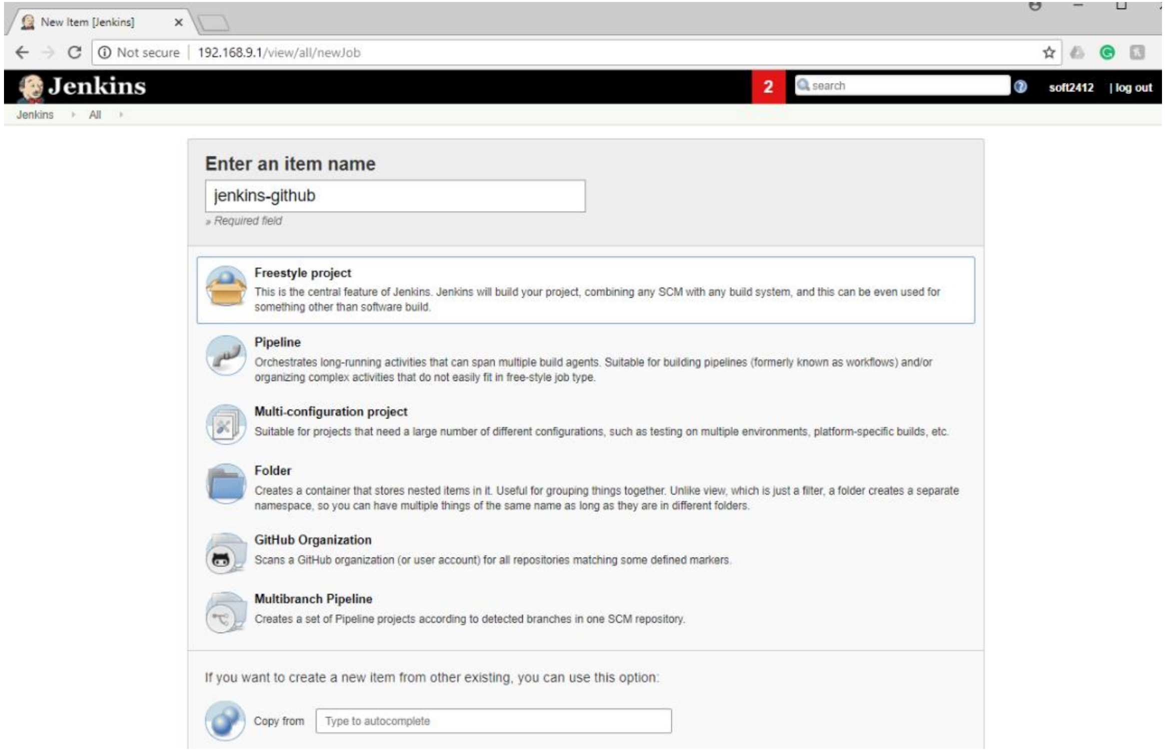


Jenkins

- "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

https://jenkins.io/

Jenkins – Integration with GitHub (1)



Enter an item name

jenkins-github

» Required field

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.

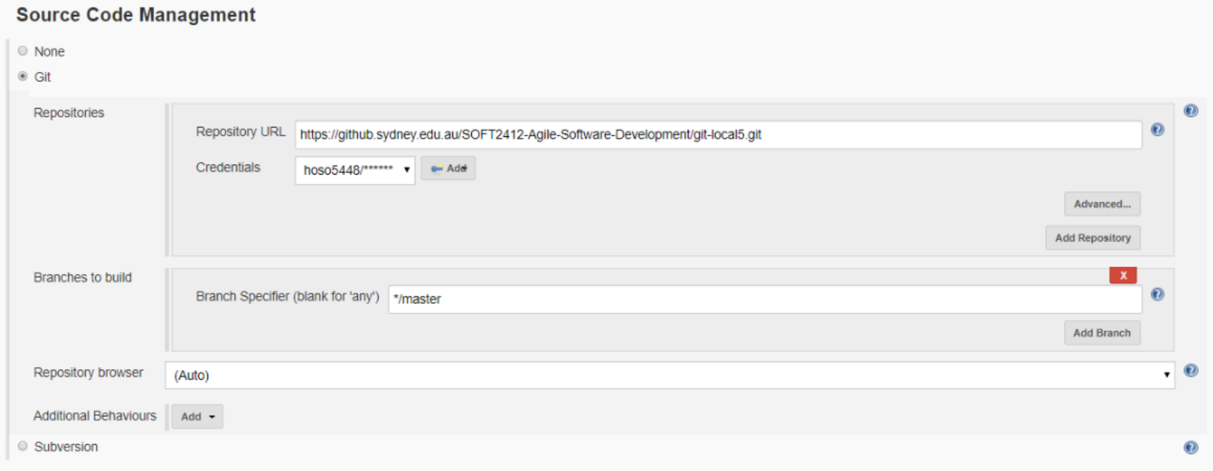
GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

If you want to create a new item from other existing, you can use this option:

Copy from: 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/****** ▾

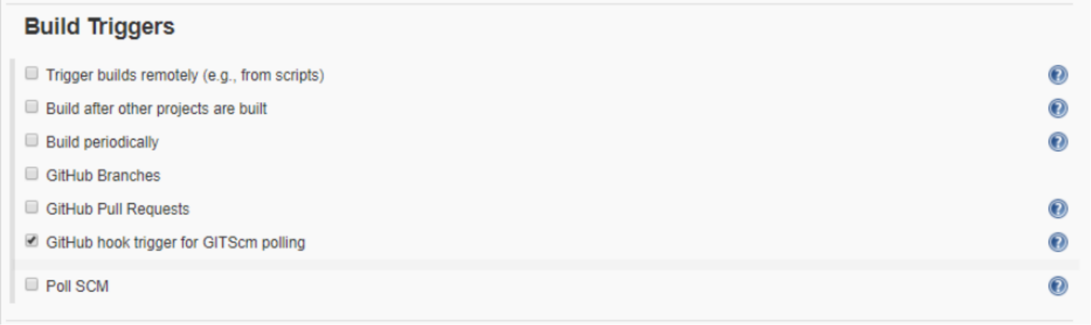
Branches to build

Branch Specifier (blank for 'any'): */master

Repository browser: (Auto) ▾

Additional Behaviours

○ 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

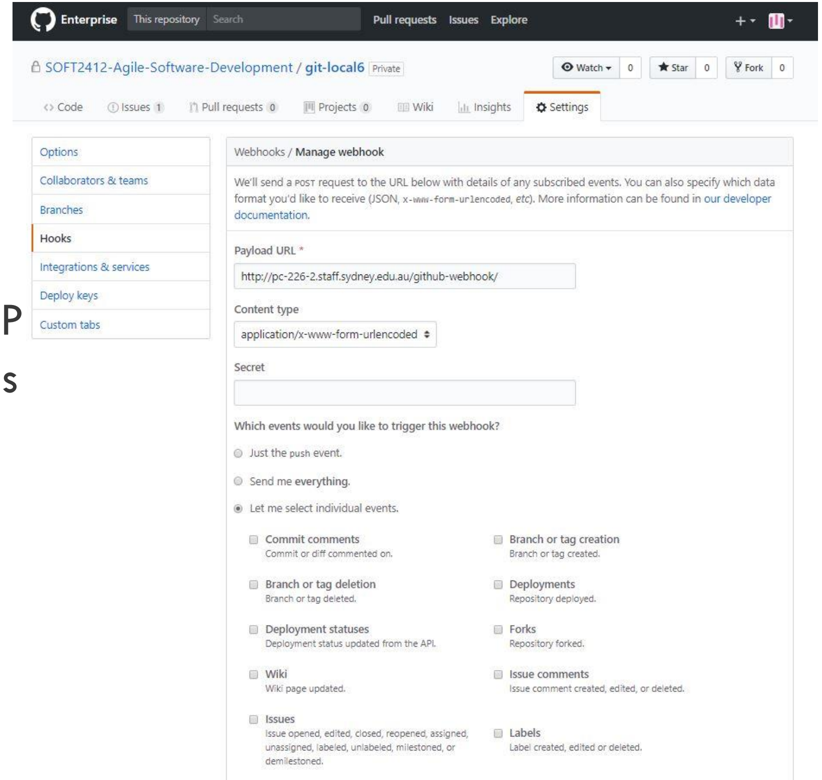
□ Poll SCM

Jenkins – Integration with GitHub (3)

- Webhooks allow to build or set up GitHub Apps which subscribe to certain events on GitHub

- When one of those 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, or even deploy to your production server



Webhooks / Manage webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.

Payload URL *

http://pc-226-2.staff.sydney.edu.au/github-webhook/

Content type

application/x-www-form-urlencoded

Secret

Which events would you like to trigger this webhook?

○ Just the push event.

○ Send me everything.

● Let me select individual events.

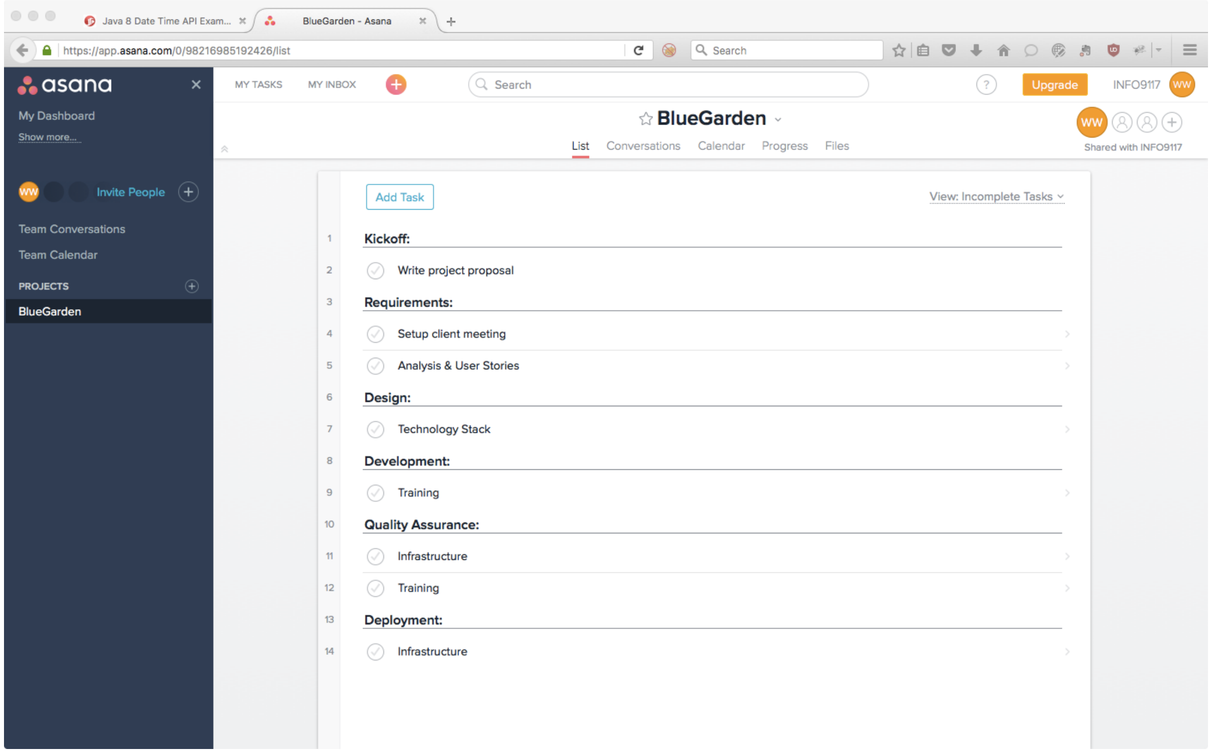
□ Commit comments Commit or diff commented on.	□ Branch or tag creation Branch or tag created.
□ Branch or tag deletion Branch or tag deleted.	□ Deployments Repository deployed.
□ Deployment statuses Deployment status updated from the API.	□ Forks Repository forked.
□ Wiki Wiki page updated.	□ Issue comments Issue comment created, edited, or deleted.
□ Issues Issue opened, edited, closed, reopened, assigned, unassigned, labeled, unlabeled, milestoned, or demilestoned.	□ Labels Label created, edited or deleted.

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



Kickoff:

- Write project proposal

Requirements:

- Setup client meeting
- Analysis & User Stories

Design:

- Technology Stack

Development:

- Training

Quality Assurance:

- Infrastructure
- Training

Deployment:

- Infrastructure

References

- Andrew Stellman, Margaret C. L. Greene 2014. Learning Agile: Understanding Scrum, XP, Lean and Kanban (1st Edition). O'Reilly, CA, USA.

- 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