

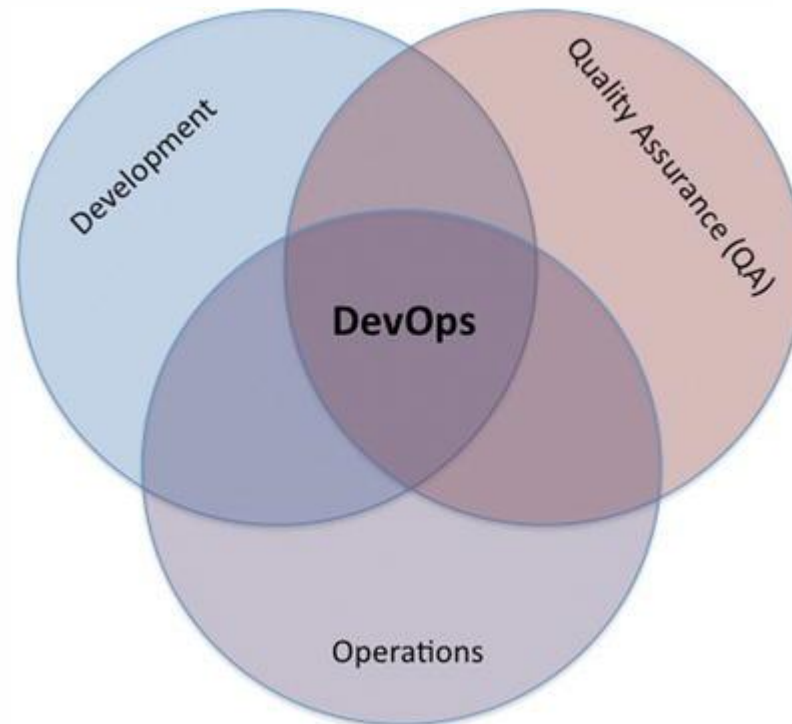


# Development Productivity Tool

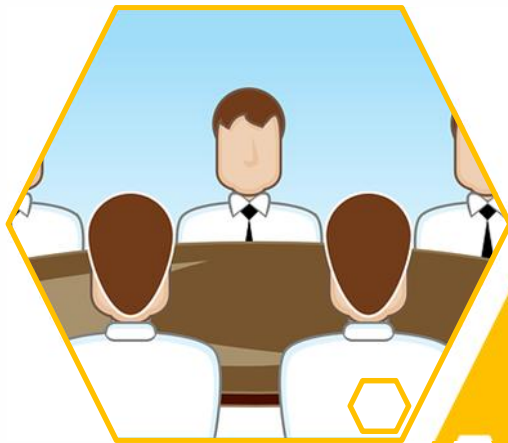
Ridi Ferdiana  
[ridi@acm.org](mailto:ridi@acm.org)  
Version 1.0.0

# DevOps Concept

- Developer & Operation: Integrating the development with daily operation.
- DevOps is more about culture and processes than it is about organization



# Why DevOps



Making sure the development is productive on the organization

Making an effort to improve the quality of software by integrating it with operation

- Excellent
- Very good
- Good
- Average
- Poor

# How to implement the DevOps

Alignment between operation and developer activity

Empowering the team with communication and collaboration tools

Measuring the effectiveness by doing continuous tracking

# Alignment model

## Engaging the Business Side

- The development should consider the business impact
- The business should see opportunity to improve by the development
- Both team agrees to understand the work load on the organization

## Sharing Information

- Sharing the status of the development with clear and continuous feedback
- Frequent release to improve business visibility

# Empowering The Team

- Collaboration workspace
  - Offline
  - Online
- Repository
  - Requirements (Files)
  - Source Codes & Binary
- Tracking
  - Tracking changes
  - Tracking bugs
  - Tracking Release

# Collaboration Workspace

- Mailing List (Email)
- Group Chat / VOIP (IM)
- Team Forum

The screenshot shows the Visual Studio Online interface for a Teams chat room. The top bar includes the Visual Studio Online logo, the user name 'Maarten Balliauw', and settings/help icons. Below the bar are tabs for 'Overview', 'Users', 'Rooms', and 'Load test'. The main chat area is titled 'GoogleAnalyticsTracker Team Room' and shows a conversation log with timestamps. The chat history includes:

- Maarten Balliauw entered the room (20:29)
- Maarten Balliauw: Just updated the code base. (20:45)
- TeamCity user: Build started (triggered by Maarten Balliauw). GoogleAnalyticsTracker - CI [master] #1.2.727, agent img-cloud-1. Includes a link to the build log. (20:48)
- Maarten Balliauw: Allright! Build is underway. (20:48)
- TeamCity user: Build successful. GoogleAnalyticsTracker - CI [master] #1.2.727, agent img-cloud-1 (Tests passed: 224; inspections total: 459, errors: 0). Includes a link to the build log. (20:56)
- Maarten Balliauw: Woohoo! 🙌 (20:57)

On the right side, there is a sidebar with 'In the room' (showing Maarten Balliauw) and 'Away' (with a 'Manage users...' link). At the bottom, there is a text input field labeled 'Post a message' with a smiley face icon for emojis.



# Demo

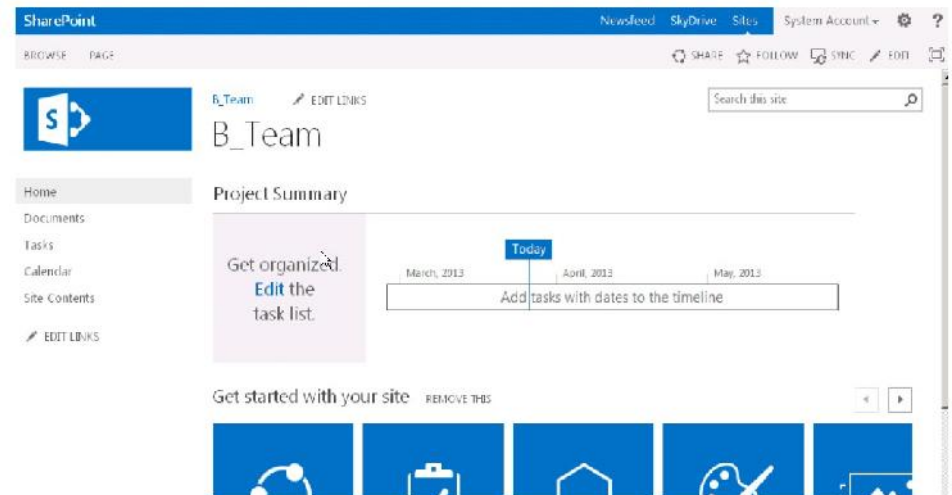
## *Setup Collaboration Workspace*

Creating Collaboration Workspace



# Repository

- Software Repository
  - Source Codes Repository
  - Binary / Production Repository
- Artifacts Repository
  - Developers log book
  - Requirements files
  - Design files
  - Backup





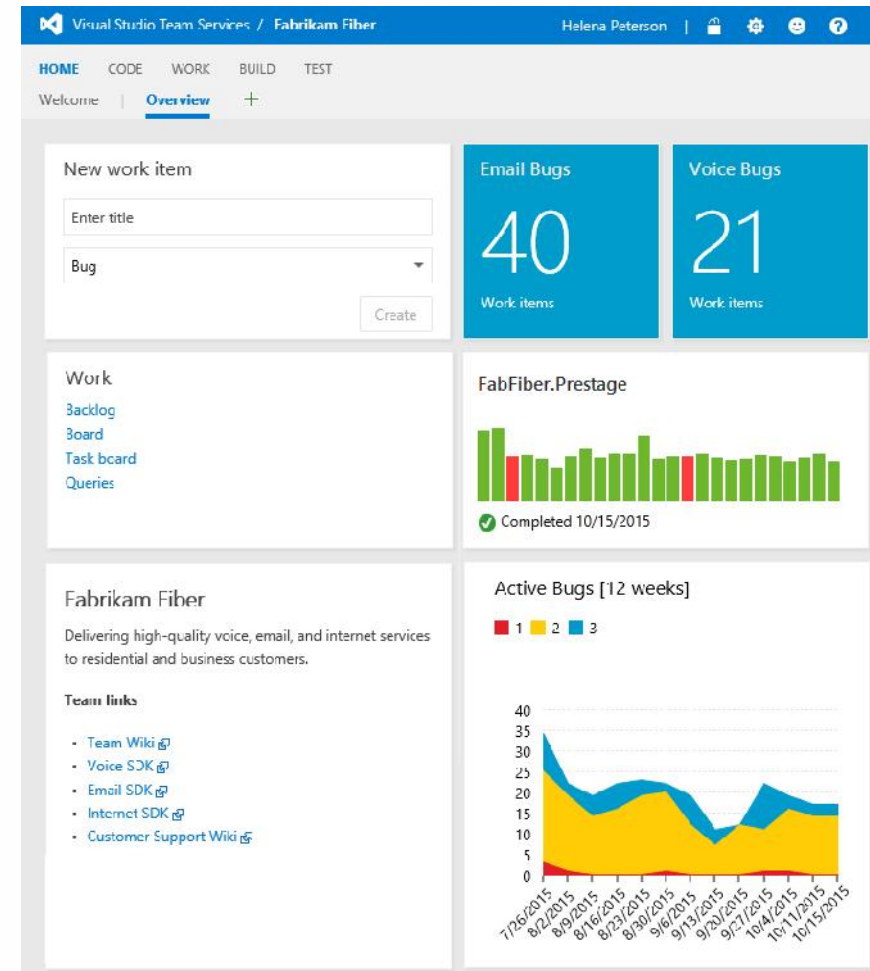
# Demo

## *Setup Repository*

Setup Repository on The Cloud

# Tracking

- Tracking the software progress
  - Continuous Build Tracking
- Tracking the software quality
  - Test plan tracking
- Tracking the software project status
  - Dashboard etc





# Video *Tracking Project*

How Dashboard and Chart Help team  
decision

# Measuring Development Productivity

Project management KPI

Architecture, analysis, and design KPI

Developer practices KPI

Software testing KPI

Release management KPI

# Project Management metrics

Backlog  
overview

Sprint  
burndown

Velocity  
report

Release  
burndown

Remaining  
work

Unplanned  
work

# Architecture & Design

*Lines of code:* This is an approximate number based on Intermediate Language (IL) code.

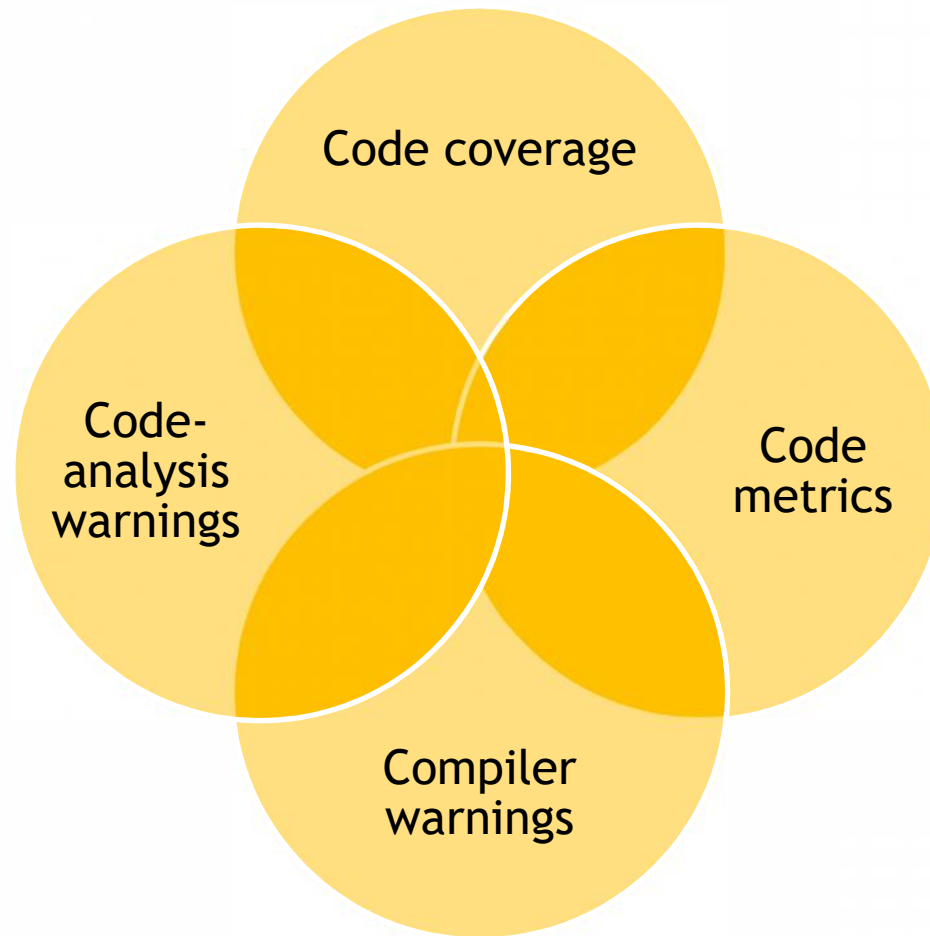
*Class coupling:* Measures the coupling to unique classes through parameters

*Depth of inheritance:* Indicates the number of class definitions that extend to the root

*Cyclomatic complexity:* Determined by calculating the number of different code paths

*Maintainability index:* An index value between 0 and 100 that represents the relative ease of maintaining the code

# Developer Practices





# Software Testing KPI

*Number of bugs per state:* How many bugs are active, resolved, or closed? Is the number of active bugs increasing and the number of resolved and closed bugs constant? If so, you need to look into how you perform your testing.

*Number of bugs sent back from testers for more information (a.k.a reactivated bugs):* A large number may indicate that communication between developers and testers must improve.

*Code coverage:* This shows how much of the code has been covered by automated unit tests. You get the value as a percentage of the entire codebase.

*Tests run results:* How are your tests performing? Do you have many failed tests? If so, what can you do to improve this?

*Percent requirements covered by test cases:* Do you write test cases for all your requirements? If not, what is the reason?

*Percent requirements covered by testing:* Do you actually run the tests for which you have test

# Release Management KPI

Number of software defects in production (the number of bugs or software defects of applications [versions] that are in production)

Percentage of successful software upgrades (excludes full installations)

Number of untested releases (not tested and signed off)

Number of urgent releases

Average costs of release, where costs most likely are based on man-hours spent

# Key points

- DevOps as a model to improve productivity on management and development
- DevOps
  - Alignment
  - Empowering
  - Measuring

# References

- Ridi Ferdiana. 2014. Application Lifecycle Management Course Module. MCT Module
- Joachim Rossberg. 2014. Beginning Application Lifecycle Management. Apress