Techniques

Contents

1. How we work
2. Unit Testing
    a. Test Driven Development
3. Toolchain (TFS)
    a. Work Item Tracking
    b. DevOps
    c. Code Quality
4. Frameworks
5. Core Technologies

How we work

What we can provide - Teams

- K&K “A-Team”: 5 full-time developers with deep specialisation in C#, XAML (WPF) & T-SQL (Microsoft SQL Server)
    - For peak throughput, support by our other K&K “Team X” possible (also 5 C# developers)
    - K&K can thus provide up to 1500 hours of development per month
- We can accommodate external developers and integrate them into the project
- We can also contract an UI/UX expert to support your wishes

Industry supported best practice

- Coding Conventions and Guidelines from Microsoft
  - Coding Conventions
    - https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions
  - Guidelines:
    - Framework Design Guidelines https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines
    - User-Experience https://docs.microsoft.com/en-us/windows/desktop/uxguide/guidelines
  - Language Features (new in version 7.3)
    - https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-7-3

Unit Testing & Test Driven Development

Unit Testing

- To ensure the quality of code, almost every component has to be runnable without the normal application context
- Must be implemented from conception, otherwise there is a risk that individual components can no longer be separated from the rest of the application and thus tested
- Unit testing as basis for proper and lasting software quality
  - ➢ Provides the basis for Test Driven Development (see next slide)
  - ➢ Generally Test Driven Development is more expensive initially, but saves resources in the long term. Even when pure Test Driven Development is unfeasible, many elements of this method of development can be adapted to fit “normal” projects.

Unit Testing: Test Driven Development

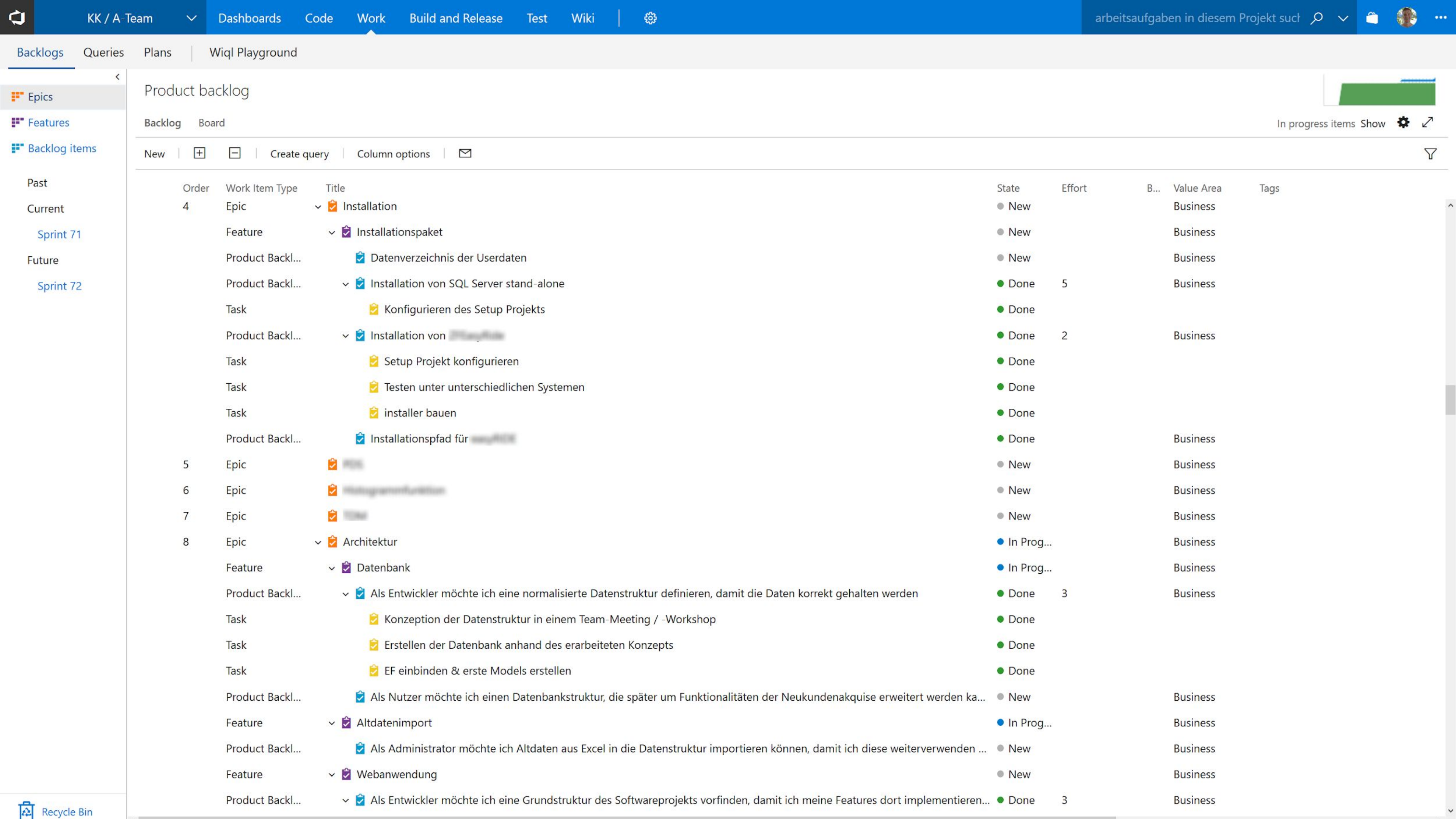
- This method of development yields earliest possible error detection and avoidance.
- Focus work on the functionality and the primary features of the software.
- Acceptance criteria and dependencies are turned into unit tests, piece by piece.
- Every change is fully automatically tested. We can recognize and correct errors while we implement user stories.

Toolchain

Toolchain

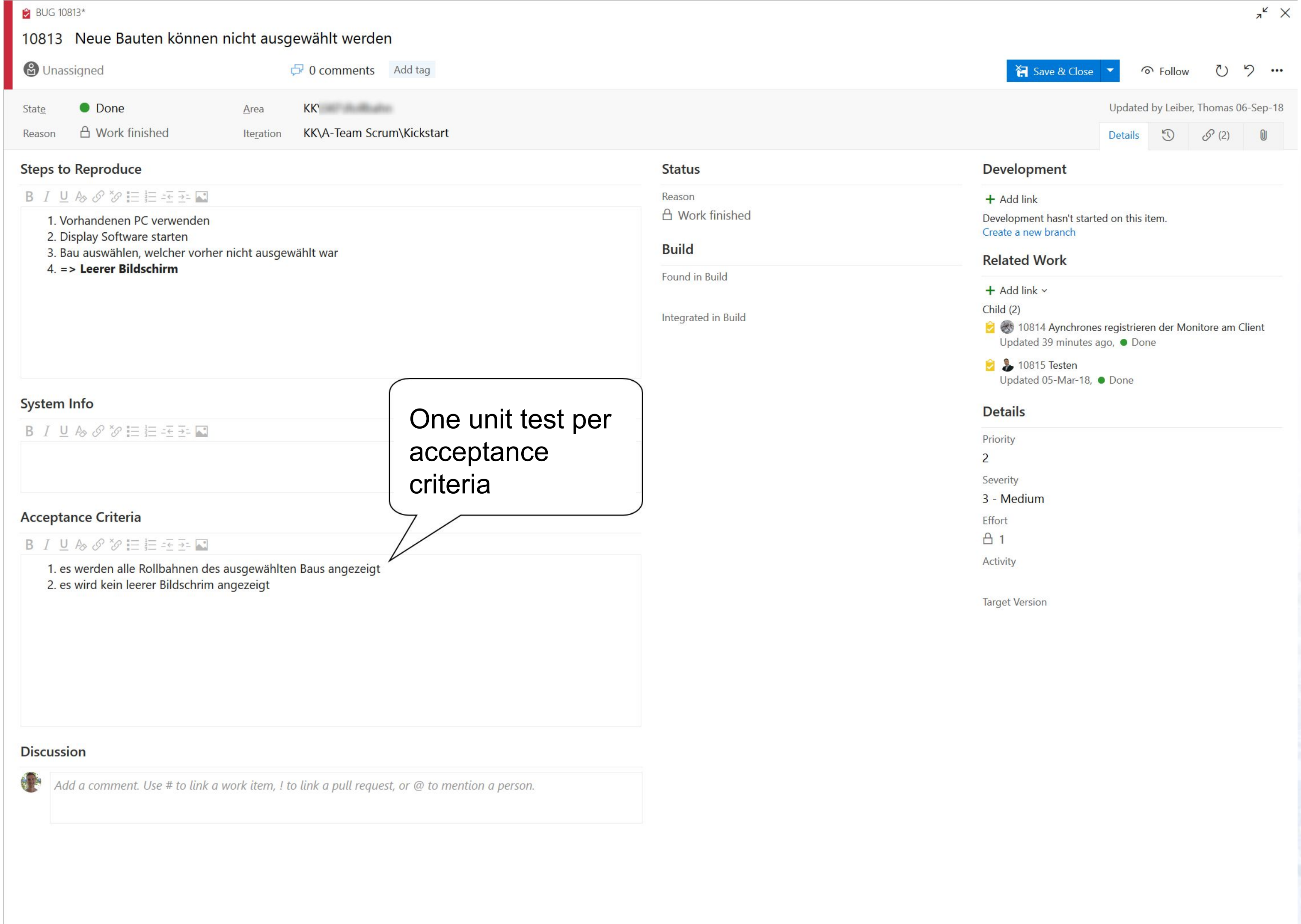
- Visual Studio (opt. Productivity Power Tools)
- Sql Server Management Studio (SSMS)
- Team Foundation Server (TFS)
  - Transparent communication and documentation at a glance
    - Shows which acceptance criteria meets the implemented user story
    - Shows all code changes to meet individual acceptance criteria
    - Shows who implemented each user story
  - Automated builds and release management
    - Is provided by us
    - Every version can be compiled and downloaded seamlessly
  - Accessible via the internet

TFS



Product backlog

Order	Work Item Type	Title	State	Effort	B...	Value Area	Tags
4	Epic	Installation	● New			Business	
	Feature	Installationspaket	● New			Business	
	Product Backl...	Datenverzeichnis der Userdaten	● New			Business	
	Product Backl...	Installation von SQL Server stand-alone	● Done	5		Business	
	Task	Konfigurieren des Setup Projekts	● Done				
	Product Backl...	Installation von [redacted]	● Done	2		Business	
	Task	Setup Projekt konfigurieren	● Done				
	Task	Testen unter unterschiedlichen Systemen	● Done				
	Task	installer bauen	● Done				
	Product Backl...	Installationspfad für [redacted]	● Done			Business	
5	Epic	[redacted]	● New			Business	
6	Epic	[redacted]	● New			Business	
7	Epic	[redacted]	● New			Business	
8	Epic	Architektur	● In Prog...			Business	
	Feature	Datenbank	● In Prog...			Business	
	Product Backl...	Als Entwickler möchte ich eine normalisierte Datenstruktur definieren, damit die Daten korrekt gehalten werden	● Done	3		Business	
	Task	Konzeption der Datenstruktur in einem Team-Meeting / -Workshop	● Done				
	Task	Erstellen der Datenbank anhand des erarbeiteten Konzepts	● Done				
	Task	EF einbinden & erste Models erstellen	● Done				
	Product Backl...	Als Nutzer möchte ich einen Datenbankstruktur, die später um Funktionalitäten der Neukundenakquise erweitert werden ka...	● New			Business	
	Feature	Altdatenimport	● In Prog...			Business	
	Product Backl...	Als Administrator möchte ich Altdaten aus Excel in die Datenstruktur importieren können, damit ich diese weiterverwenden ...	● New			Business	
	Feature	Webanwendung	● New			Business	
	Product Backl...	Als Entwickler möchte ich eine Grundstruktur des Softwareprojekts vorfinden, damit ich meine Features dort implementieren...	● Done	3		Business	



BUG 10813*

10813 Neue Bauten können nicht ausgewählt werden

Unassigned · 0 comments · Add tag

State: Done

Reason: Work finished

Area: KK\[redacted]

Iteration: KK\A-Team Scrum\Kickstart

Updated by Leiber, Thomas 06-Sep-18

Steps to Reproduce

1. Vorhandenen PC verwenden
2. Display Software starten
3. Bau auswählen, welcher vorher nicht ausgewählt war
4. **=> Leerer Bildschirm**

System Info

Acceptance Criteria

1. es werden alle Rollbahnen des ausgewählten Baus angezeigt
2. es wird kein leerer Bildschrim angezeigt

One unit test per acceptance criteria

Discussion

Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.

Status

Reason: Work finished

Build

Found in Build

Integrated in Build

Development

+ Add link

Development hasn't started on this item.

Create a new branch

Related Work

+ Add link

Child (2)

- 10814 Aynchrones registrieren der Monitore am Client
  Updated 39 minutes ago, ● Done
- 10815 Testen
  Updated 05-Mar-18, ● Done

Details

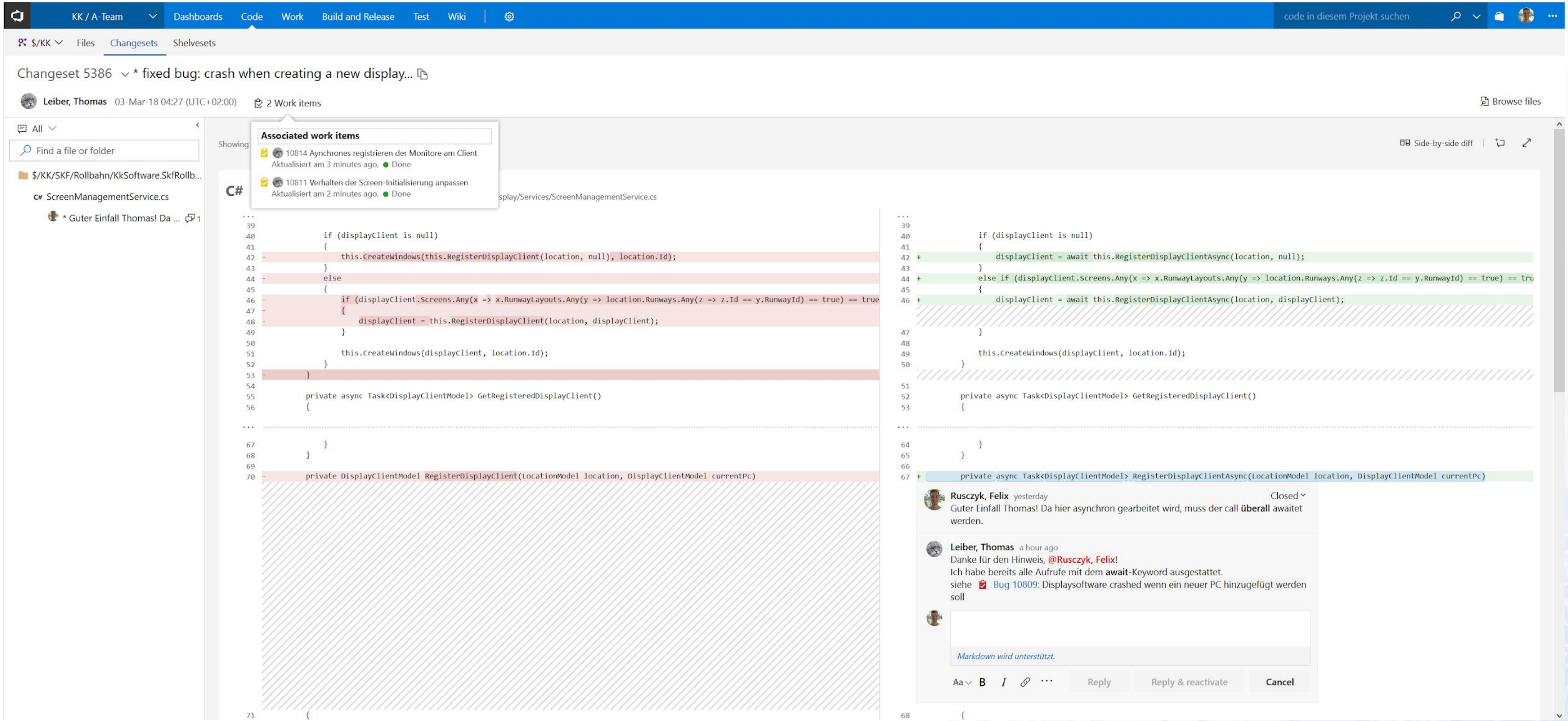
Priority: 2

Severity: 3 - Medium

Effort: 1

Activity

Target Version



Changeset 5386 * fixed bug: crash when creating a new display...

Leiber, Thomas 03-Mar-18 04:27 (UTC+02:00) 2 Work items

$/KK/SKF/Rollbahn/KkSoftware.SkfRollb...

ScreenManagementService.cs

* Guter Einfall Thomas! Da ... 1

Showing

Associated work items

10814 Aynchrones registrieren der Monitore am Client
Aktualisiert am 3 minutes ago, ● Done

10811 Verhalten der Screen-Initialisierung anpassen
Aktualisiert am 2 minutes ago, ● Done

...splay/Services/ScreenManagementService.cs

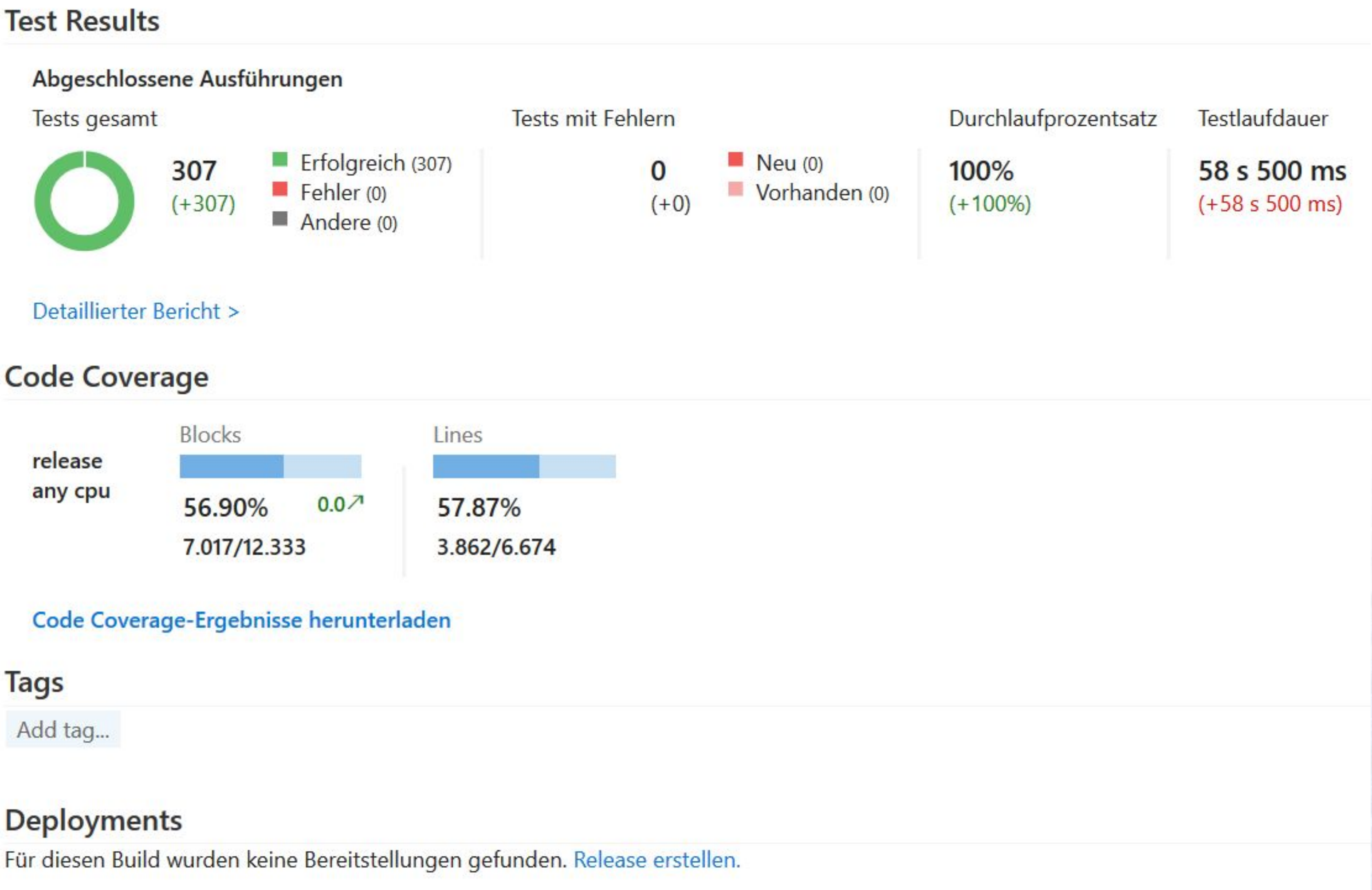
```
            if (displayClient is null)
            {
-               this.CreateWindows(this.RegisterDisplayClient(location, null), location.Id);
            }
-           else
            {
-               if (displayClient.Screens.Any(x => x.RunwayLayouts.Any(y => location.Runways.Any(z => z.Id == y.RunwayId) == true) == true
-               {
-                   displayClient = this.RegisterDisplayClient(location, displayClient);
                }

                this.CreateWindows(displayClient, location.Id);
            }
-       }

        private async Task<DisplayClientModel> GetRegisteredDisplayClient()
        {
...
            }
        }

-       private DisplayClientModel RegisterDisplayClient(LocationModel location, DisplayClientModel currentPc)
        {
```

```
            if (displayClient is null)
            {
+               displayClient = await this.RegisterDisplayClientAsync(location, null);
            }
+           else if (displayClient.Screens.Any(x => x.RunwayLayouts.Any(y => location.Runways.Any(z => z.Id == y.RunwayId) == true) == tru
            {
+               displayClient = await this.RegisterDisplayClientAsync(location, displayClient);
            }

            this.CreateWindows(displayClient, location.Id);
        }

        private async Task<DisplayClientModel> GetRegisteredDisplayClient()
        {
...
            }
        }

+       private async Task<DisplayClientModel> RegisterDisplayClientAsync(LocationModel location, DisplayClientModel currentPc)
        {
```

Rusczyk, Felix yesterday Closed
Guter Einfall Thomas! Da hier asynchron gearbeitet wird, muss der call **überall** awaitet werden.

Leiber, Thomas a hour ago
Danke für den Hinweis, @Rusczyk, Felix!
Ich habe bereits alle Aufrufe mit dem **await**-Keyword ausgestattet.
siehe Bug 10809: Displaysoftware crashed wenn ein neuer PC hinzugefügt werden soll

Markdown wird unterstützt.



Test Results

Abgeschlossene Ausführungen

Tests gesamt

307
(+307)

🟩 Erfolgreich (307)
🟥 Fehler (0)
⬛ Andere (0)

Tests mit Fehlern

0
(+0)

🟥 Neu (0)
🟥 Vorhanden (0)

Durchlaufprozentsatz

100%
(+100%)

Testlaufdauer

58 s 500 ms
(+58 s 500 ms)

Detaillierter Bericht >

Code Coverage

	Blocks		Lines
release any cpu	56.90%	0.0↗	57.87%
	7.017/12.333		3.862/6.674

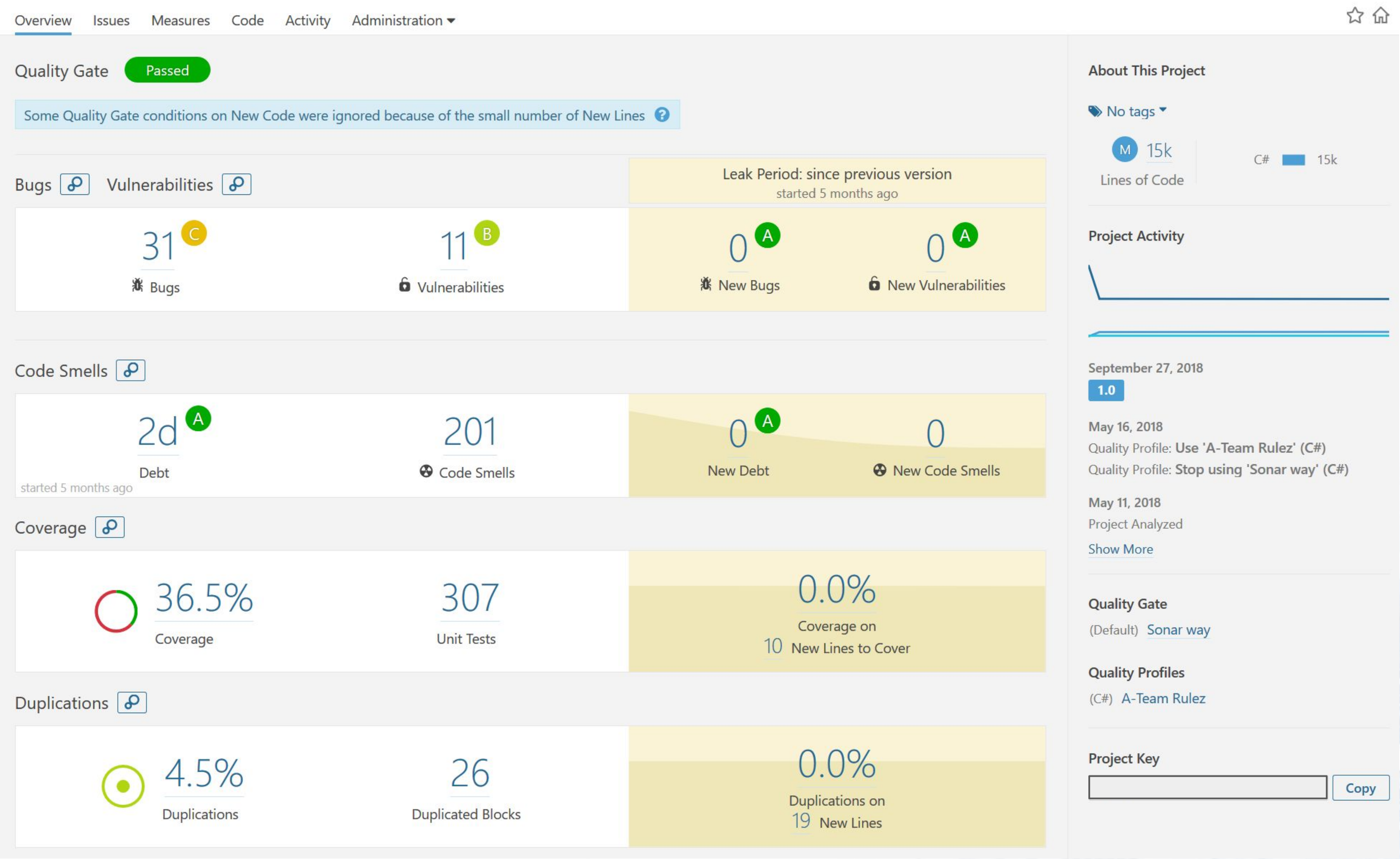
Code Coverage-Ergebnisse herunterladen

Tags

Add tag...

Deployments

Für diesen Build wurden keine Bereitstellungen gefunden. Release erstellen.



Quality Gate **Passed**

Some Quality Gate conditions on New Code were ignored because of the small number of New Lines

Bugs

Vulnerabilities

Leak Period: since previous version
started 5 months ago

31 C Bugs	11 B Vulnerabilities	0 A New Bugs	0 A New Vulnerabilities

Code Smells

2d A Debt started 5 months ago	201 Code Smells	0 A New Debt	0 New Code Smells

Coverage

36.5% Coverage	307 Unit Tests	0.0% Coverage on 10 New Lines to Cover

Duplications

4.5% Duplications	26 Duplicated Blocks	0.0% Duplications on 19 New Lines

About This Project

No tags ▾

M 15k
Lines of Code

C# 15k

Project Activity

September 27, 2018
1.0

May 16, 2018
Quality Profile: Use 'A-Team Rulez' (C#)
Quality Profile: Stop using 'Sonar way' (C#)

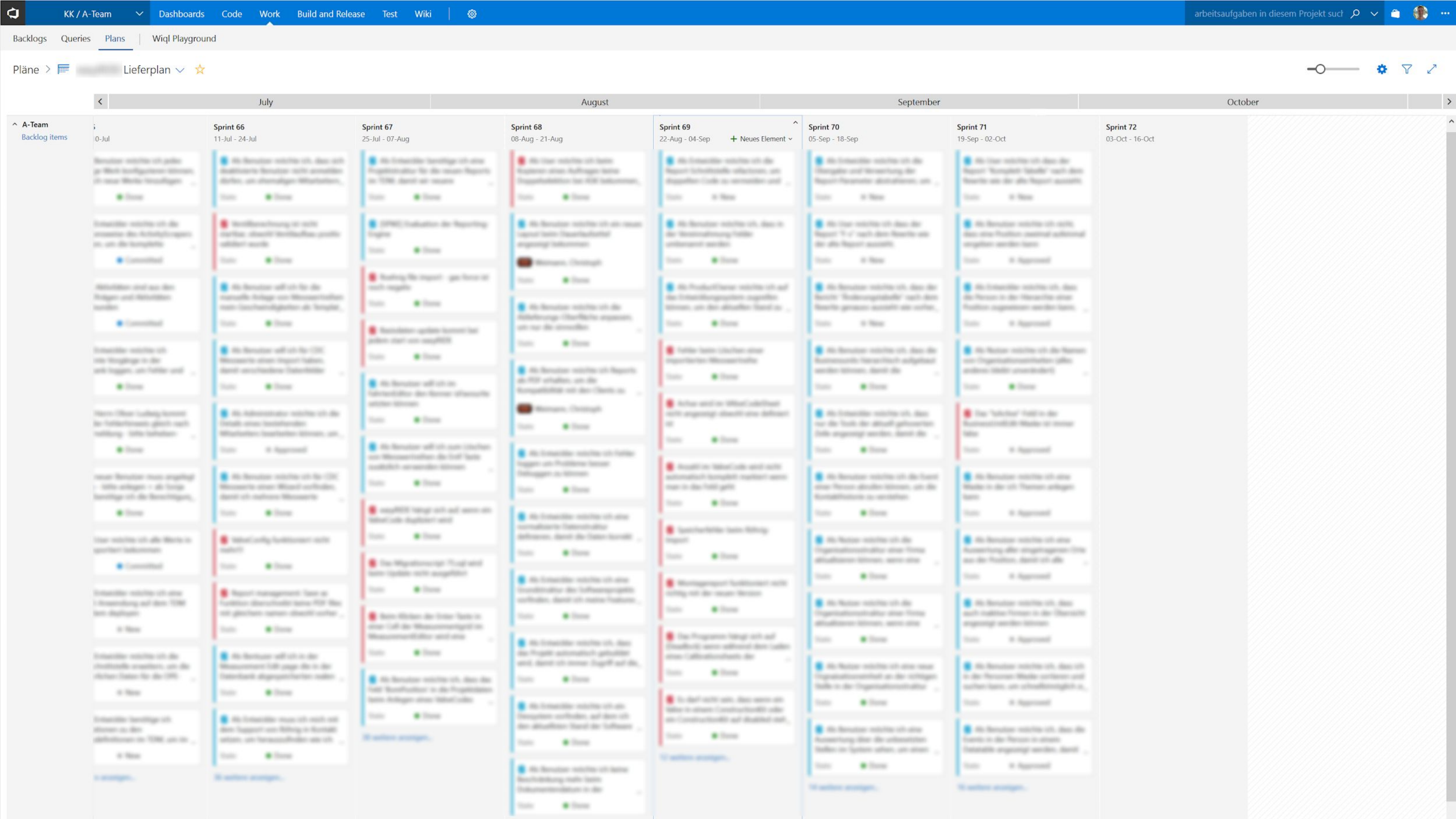
May 11, 2018
Project Analyzed

Show More

Quality Gate
(Default) Sonar way

Quality Profiles
(C#) A-Team Rulez

Project Key



0-Jul	Sprint 66	Sprint 67	Sprint 68	Sprint 69	Sprint 70	Sprint 71	Sprint 72
	11-Jul - 24-Jul	25-Jul - 07-Aug	08-Aug - 21-Aug	22-Aug - 04-Sep	05-Sep - 18-Sep	19-Sep - 02-Oct	03-Oct - 16-Oct

Frameworks and Techniques

Current frameworks and Techniques

Use of best practice guidelines ⇒ we do not reinvent the wheel.

- PRISM Framework
- Repository in conjunction with Unit-of-Work design pattern
- Dependency Injection
- Façades
- Unit Testing
- Entity Framework

PRISM

- PRISM is a powerful open source MVVM library
- Glossary
  - **MVVM** = loose coupling between Model, View and ViewModel
  - **M**odel = data model (intermediate layer to represent the data from the database)
  - **V**iew = display objects (graphical user interface & visual representation of ViewModel)
  - **V**iew**M**odel = state of the interface (and link between model and view)
- Benefits
  - Supporting design patterns that embody important principles of architectural design (e.g. “Separation of Concerns”), PRISM helps developers design and build loosely-coupled components that can be developed independently, but jet can be easily and seamlessly integrated into the overall application.

Repository and Unit-of-Work pattern

- Repository pattern in conjunction with unit-of-work pattern
  - Glossary
    - Repository ⇒ Representation and collection of database operations for business logic
    - Unit-of-Work ⇒ Collection of repositories within a mask
  - Benefits
    - Data is created, edited, and deleted centrally in one place
    - Prevention of duplicate code as this layer can be accessed from anywhere
- ⇒ We are writing future-oriented and maintainable code!

Dependency Injection

- Is a technique to minimize fixed references
- Increases the testability due to hiding complex code fragments behind simplified interfaces => code lines are not needlessly tested twice
- The library “Unity” is used as our Dependency Injector
- Centralized dependency management, which “inserts” the appropriate code fragment (e.g. Service, Factory, or Façade, etc.) into objects

Façade

- Hides complex code fragments behind a simple “facade”
  - Code irrelevant to the developer is hidden
    ⇒ improves readability of code
    ⇒ especially important if the code is modified by several developers

Frontend frameworks

- Xceed Toolkit
  - See Live-Explorer for Demo: https://github.com/xceedsoftware/wpftoolkit/releases
  - AvalonDock
    - Window management for complex applications
  - User Controls
    - A variety of components for editing, creating, and deleting data
    - Easily expandable

Core Technologies

C# common toolsets

- Nuget Package Manager to provide specially designed source code libraries
- Encryption and authentication of communications to ensure data integrity and security
- High-performance applications with multi-threaded programming

- “MVP” Concept
    - Minimum Viable Product
- Design patterns
    - Dependency Injection (Unity, …)
    - Asynchronous Programming Model
    - Inversion of Control
    - …
- We support a wide range of database management systems. Depending on project requirements Microsoft SQL Server or SQLite are preferred.
- Unit Testing
    - NUnit
    - Moq

Frontend Technology

Desktop-Client

- Windows Presentation Foundation (WPF)
- MVVM Light Toolkit or PRISM, depending on the scope of the project.
- OxyPlot to generate graphs and diagrams
- Xceed Toolkit to support your UI needs.
- AvalonDock to support multi window applications

Web-Client

- Backend
  - ASP.NET MVC 6 or ASP.NET Core MVC depending on platform
- Frontend
  - Current web html5 technology
  - Typescript
  - Bootstrap 4
  - Angular

Any questions? Please contact us!

We are committed to your request and will be happy to answer detailed questions promptly. We look forward to your inquiry.

I am your contact person:

Laura Köpl
Head of Marketing and Sales

Phone:	09382 / 3102-241
Mail:	koepl@kk-software.de

Protect your interests:

Develop excellent products with our team at K&K Software AG using state-of-the-art agile methods, now.