



CYBERTEC

ZUCCHETTI

Our values

PURPOSE

Cybertec was founded in 1991 from the intersection of technology and a passion for the supply chain, with the aim of aligning operational decisions with business objectives.

VISION

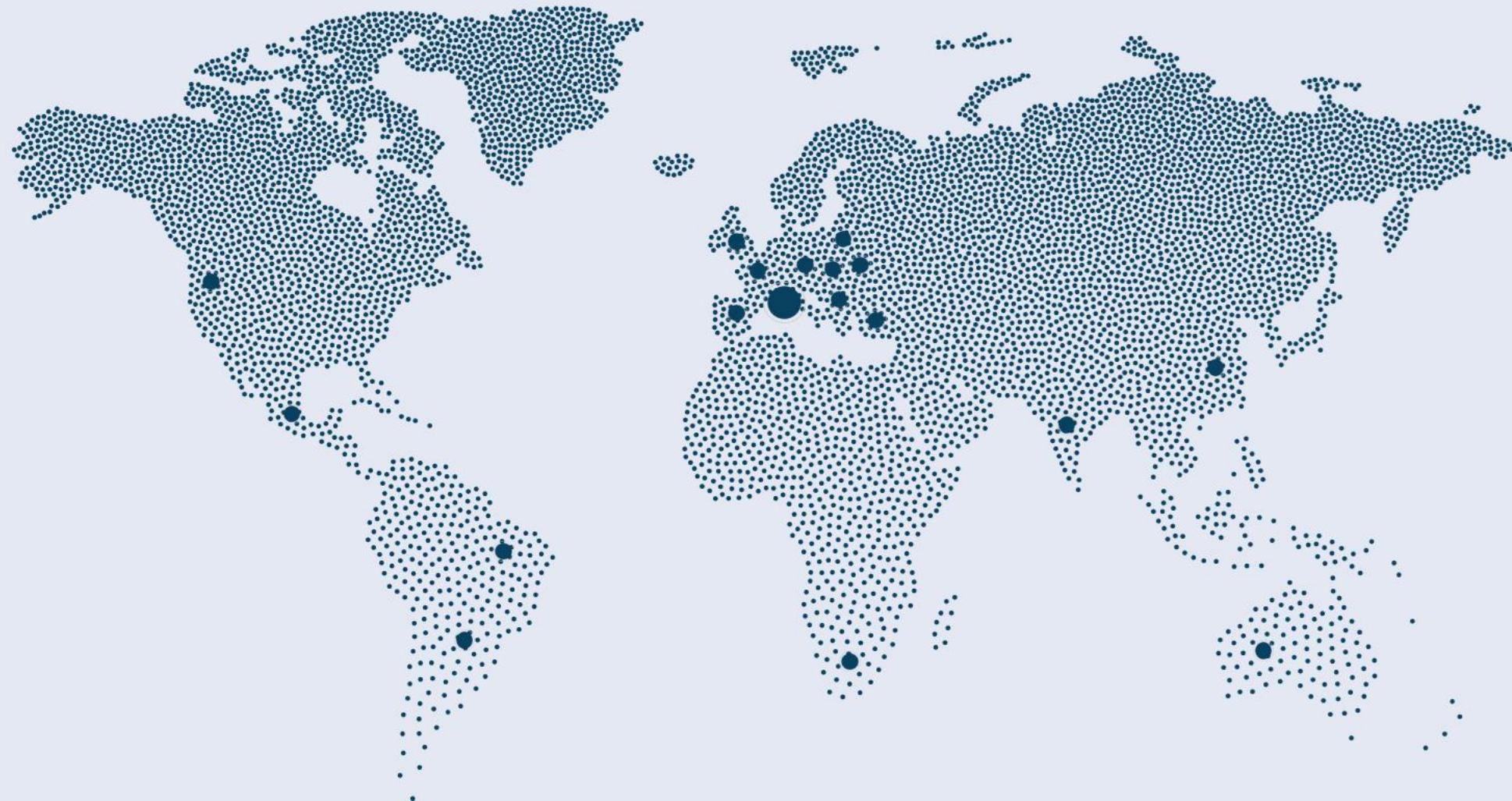
For over 30 years, we've been putting innovation at the service of the supply chain, from planning and scheduling (APS) to production process execution (MES/MOM). We develop intuitive and constantly updated software to manage production processes.

MISSION

Helping manufacturing companies continuously improve their performance and the quality of their operations.

International presence

APS/MES/MOM worldwide projects





FROM 2020 PART OF THE GROUP



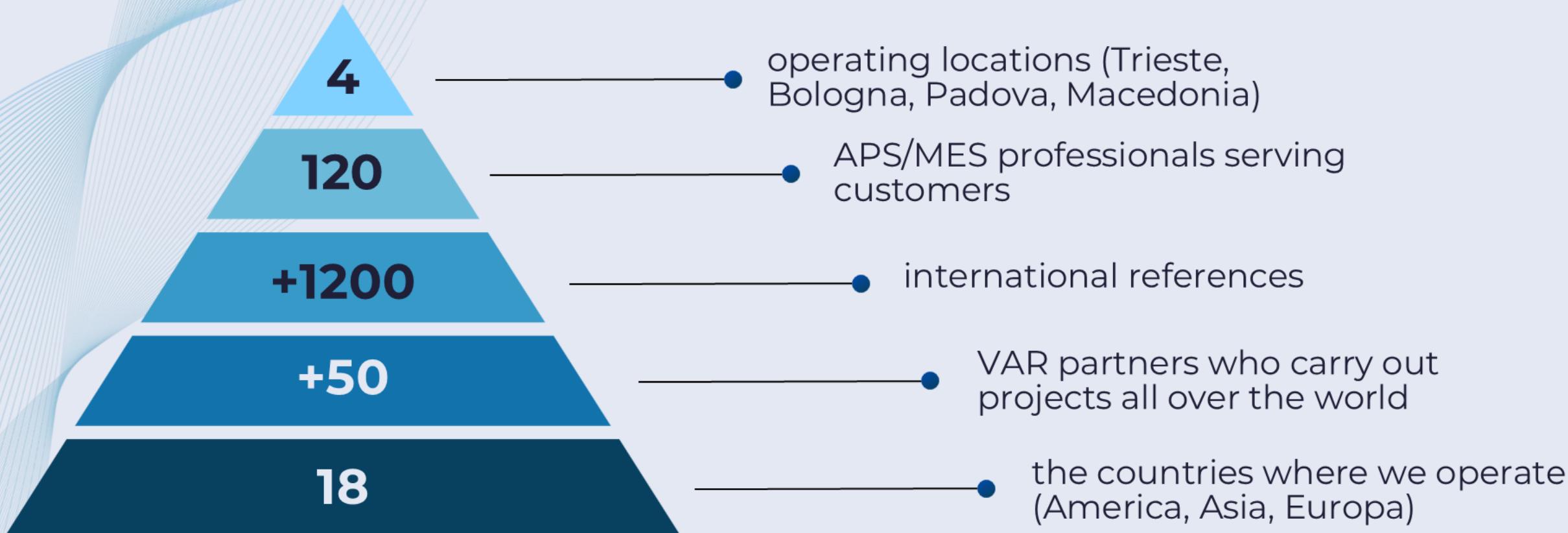
CYBERTEC

ZUCCHETTI



ZUCCHETTI

9000 people
2.4 billion in turnover
700.000 customers
Global presence



SOME CYBERTEC REFERENCES IN VARIOUS SECTORS

ETO



Zoppas Industries



MTO



EMERSON



RO



Lima Corporate
Orthopaedic motion

ATO



Nutkao

Morato
PANE&IDEE



MTS

MARCOLIN
EYEWEAR



THÉLIOS

VIMAR
energia positiva

Reckitt Benckiser



Two Business Units dedicated to the development of our software



Opera MES

Software for real-time production
management and control

CYBERPLAN

Production planning software

Our Workflow: Engineering & DevOps

● Agenda

- Sprint Planning
- Development
- Pipelines (CI/CD)
- QA
- Evaluation
- Release
- Lesson Learned
- Questions

Sprint Planning

1. Sprint Creation in Jira Software

- The PM initiates the sprint creation in Jira Software, the tool used to manage Agile workflows
- In our workflow, one sprint has a duration of two weeks
- The week before the current sprint ends, the Project Manager (PM) defines the goals and priorities for the upcoming sprint and assigns Jira work items to developers



Sprint Planning

2. Daily Standup Meeting

- The Standup Meeting has a fixed duration of 15 minutes
- Each team member answers:
 1. What did I do yesterday?
 2. What will I do today?
 3. Are there any blockers?
- The team realigns on priorities or dependencies that may have shifted



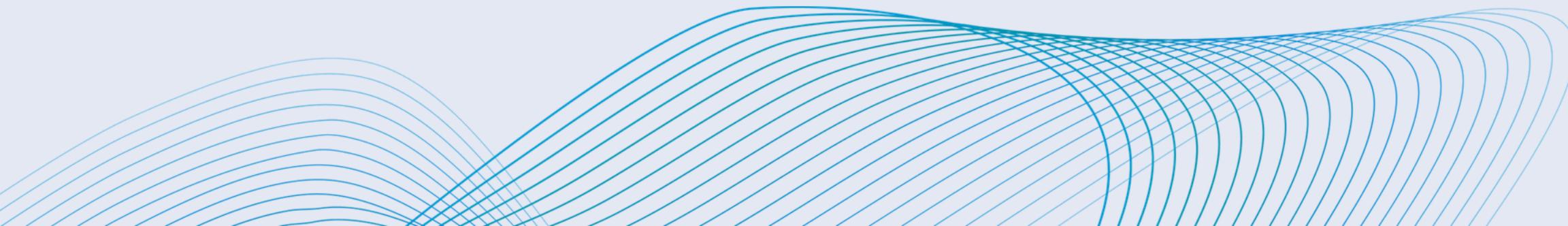
Sprint Planning

3. Jira work item workflow

- Work items follow a structured flow:

To Do → Ready → In Progress → QA → Evaluation → QA Release → Done

- This ensures transparency and traceability throughout the development lifecycle
- Each developer is responsible for keeping their Jira board updated to reflect the current status of their tasks



Sprint Planning

4. Work item Types

- We classify work items into four distinct categories:



- Apply critical thinking:
Is it a bug, missing feature,
or misuse?
- If a bug: identify when it was
introduced and which task
caused it
- Focus on root cause, not
blame; ensure fixes prevent
regressions

Lorenzo Airey 8 July 2025 at 12:58 (edited)

Source Inspection

E' un bug ? Si.

E' un problema di codice ? Si.

Esiste un workaround ? No.

I problemi individuati sono 2:

- Il salvataggio dei dati su Cyberplan, per le connessioni di tipo "native", rimuove le virgolette dai dati inviati dal Connect.
- Il connect aggiunge alle connessioni di tipo Oracle un altro paio di virgolette. (Il fatto che Oracle native rimuova un paio di virgolette è dovuto al problema 1.)

i Entrambi i problemi sono presenti sin dalla nascita del Connect.
Il problema 1 risale al ticket CYBWEB-8102: Retrocompatibilità api integrazione del plant CLOSED
Il problema 2 risali ai ticket CONNECT-13: Connect invia batch malformate CLOSED e CYBWEB-2211: Creare un programma per inviare / ricevere dati verso le API di integrazione CLOSED per le versioni 0.8.2 di Cyberplan web.

Problema 1.
All'interno del file integrationService.ts precisamente dentro nativeBatchProcess a riga 358 viene chiamata la funzione CSVParseFactory.getParser, all'interno della funzione si decide se utilizzare il csvParserV3 (che è stato costruito per l'applicativo) nel caso i dati non siano RFCCompliant oppure utilizzare il parser della libreria che è RFCCompliant. (Caso introdotto con il CYBWEB-19001: connect 1.13.0 - protocolversion 4 errore nell'invio di stringhe con le virgolette " RESOLVED)

Vi sono possibili strade, la prima è rendere i dati inviati dal Connect RFCCompliant, in questo modo si può utilizzare direttamente il parser introdotto tramite la libreria, e si risolverebbe in modo coerente questa differenza tra salvataggio Native e JSON.

Sprint Planning

5. Work item Types

- We classify work items into four distinct categories:



Analysis

- Consider project constraints and requirements
- Research existing solutions (avoid reinventing the wheel)
- Propose secure, backward-compatible solutions based on user cases

Lorenzo Airey 30 September 2025 at 11:51 (edited)

Il Connect non ritorna dati RFC Compliant, in particolare si parla di RFC 4180.

Standard

Per essere definito RFC Compliant con l'RFC 4180, un elemento deve possedere le seguenti caratteristiche:

- Ogni record è una riga, terminata da **CRLF** (`\r\n`). (Molti parser accettano anche un LF singolo per compatibilità, ma RFC richiede CRLF.)
- I campi sono separati da una **virgola** (`,`). Separatori alternativi non sono RFC.
- I campi **possono** essere racchiusi tra doppi apici (`"`).
- Se un campo contiene **virgolette**, CR, LF o la virgola, **dove** essere racchiuso tra doppi apici.
- Per includere una doppia virgola all'interno di campo racchiuso, si usa la sequenza `""` (due doppi apici).
- La prima riga **può** essere un header con nomi campo (è opzionale).
- I campi possono essere vuoti (es. `a,,c` ha un campo vuoto intermedio).
- Lo standard non specifica una codifica. Per praticità **UTF-8** è la scelta raccomandata.
- Deve rifiutare (o segnalare) record con **virgolette aperte non chiuse** o escape non validi.

Possibili soluzioni

Per rendere il Connect compliant con lo standard vi sono 2 strade:

1. Introdurre una libreria che si occupi del parsing dei dati di tipo csv.
2. Scrivere il codice da 0 per garantire maggior controllo.

La strada 1 è ovviamente la meno dispendiosa e la più manutenibile, dato che per eventuali variazioni od evoluzioni dello standard se ne occuperebbero gli sviluppatori della libreria (purché sia manutenuta).

Per il punto 1, esiste una libreria disponibile tramite NuGet (per C#) chiamata [CsvHelper](#).

A library for reading and writing CSV files. Extremely fast, flexible, and easy to use. Supports reading and writing of custom class objects.

Sprint Planning

6. Work item Types

- We classify work items into four distinct categories:



Bug

- Analyze why the issue occurred
- Reuse existing solutions if possible
- Maintain backward compatibility

Spaces / 🚩 CyberPlan web / 🔍 CYBWEB-19331 / ❌ CYBWEB-16866

Where Used: alcuni legami non vengono disegnati

+ @

Key details

Parent CYBWEB-19331 [2024INT01587] Bug CyberPlan Web 2025

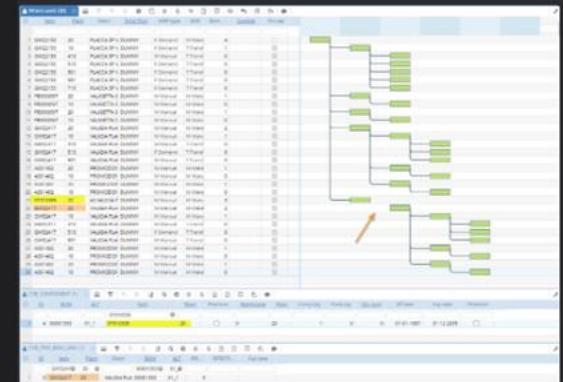
Sprint Add sprint +22

Description

Caricare il database allegato qui:

1

Aprire il quadrante Where Used per il materiale 18062356 plant 28 (quello segnalato nel ticket CYBWEB-15748: Verifica legami non disegnati su altri quadrianti CLOSED) Scrolling si osserva che i legami degli item al record 172, 238 e 363 non vengono disegnati. Si esclude il problema di paginazione in quanto la cosa si riproduce anche con un materiale con meno materiali assemblati, esempio GW22153 plant 28



da F12 si rileva in console:

Sprint Planning

7. Work item Types

- We classify work items into four distinct categories:



Task

- Implement new functionality, reusing/refactoring existing code where possible
- Apply clean code principles
- Include unit tests (e.g., Jest for TypeScript, Boost for C++)

The screenshot shows a work item details page in CyberPlan web. The title of the work item is "Esporta in edit mode la possibilità di associare uno script di transazione al doppio click sull'riga".
Key details:

- Parent: CYBWEB-19332 [2024INT01584] Miglioramenti CyberPlan Web 2025
- Sprint: Add sprint +13

Description:

RSP ha associato delle transazioni custom al doppio click di una riga. Questa cosa è impostata a livello di settings ma non è fruibile lato GUI, né in edit mode né dal wks di gestione settings.

Technical description:

None

Attachments: 3

- image-202408... 819.png (22 Aug 2024, 03:18 pm)
- image-202408... 200.png (22 Aug 2024, 03:12 pm)
- cambiaTransaz... (1).cws (09 Oct 2023, 04:52 pm)

Subtasks:

Add subtask

Sprint Planning

8. Jira Board example

The screenshot shows a Jira board titled "Development" with the "Active sprints" tab selected. The board is organized into columns: TO DO, READY, IN CLARIFICATION, SUSPENDED, IN PROGRESS, QA, EVALUATION, and DONE. Each column contains work items represented by cards.

- TO DO:** Contains one card: "User Dev board".
- READY:** Contains one card: "Cyberplan WEB".
- IN CLARIFICATION:** Contains one card: "User Release Manag...".
- SUSPENDED:** Contains two cards:
 - "Gestione tipi numerici null in form cws" (Status: V1.13.3, Due: 18/07/25)
 - "Filtri colonna non funzionanti con carattere apice ('')" (Status: V1.12.11, Due: 30/04/25)
- IN PROGRESS:** Contains one card: "Versione 1.13.1 non rimangono salvati gli ordinamenti" (Status: V1.13.3, Due: 17/10/25).
- QA:** Contains three cards:
 - "Porting 1.12 - Ricalcola data fine con actual date..." (Status: V1.12.11, Due: 21/11/25)
 - "Risultati ricerca browser considerano anch..." (Status: V1.13.3, Due: 30/09/25)
 - "[Penetration Test] Manuale con libreria JQuery vulnerabile" (Status: V1.13.3, Due: 30/09/25)
- EVALUATION:** Contains one card: "[SonarQube] Riduzione Critical [2024INT01584] REFACT..." (Status: V1.14.0, Due: 01/10/25).
- DONE:** Contains one card: "CYBWEB-19278" (Status: CYBWEB-19278, Due: 01/10/25).

The sidebar on the left shows the "CyberPlan web" space and its boards, including "Development", "User Dev board", "Cyberplan WEB", and "User Release Manag...". The top navigation bar includes links for Backlog, Releases, Timeline, Active sprints, Components, List, Summary, Calendar, Reports, All work, and ScriptRunner Enhanced Search.

Development

1. Technology Stack

Languages and Architecture

- TypeScript 
- JavaScript 
- React 
- Node.js 
- PostgreSQL 
- C# 
- C++ 

Package Management

- Yarn 

Version Control

- Git with Bitbucket 

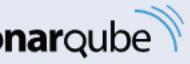
CI/CD

- Bamboo 

Test

- Jest 

Static Code Analysis

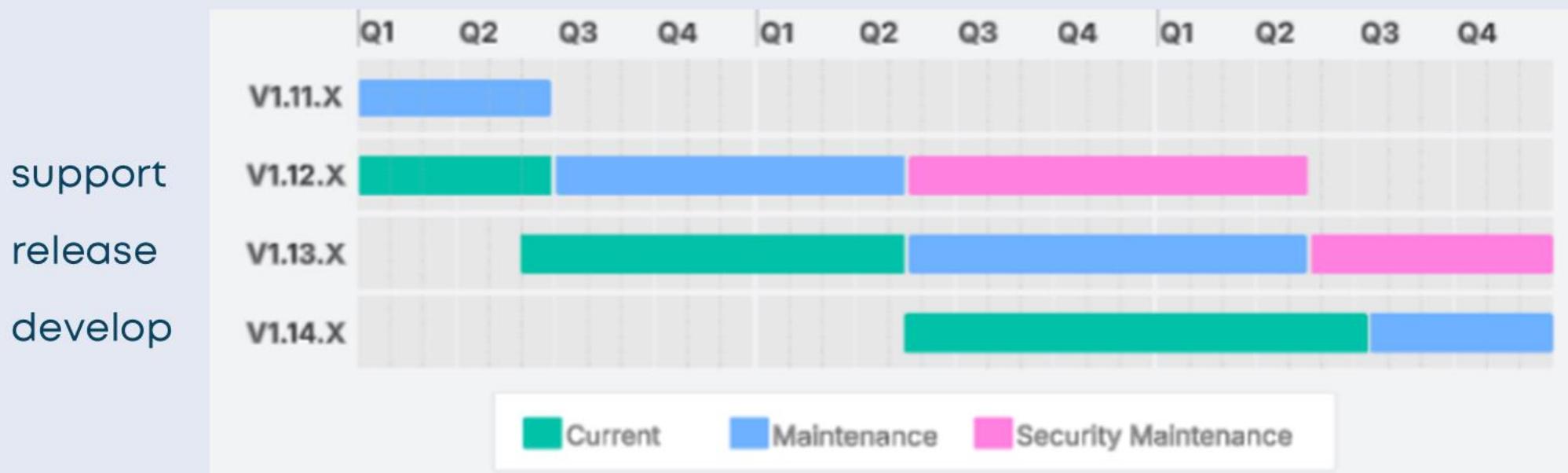
- SonarQube 

Development

2. Repositories

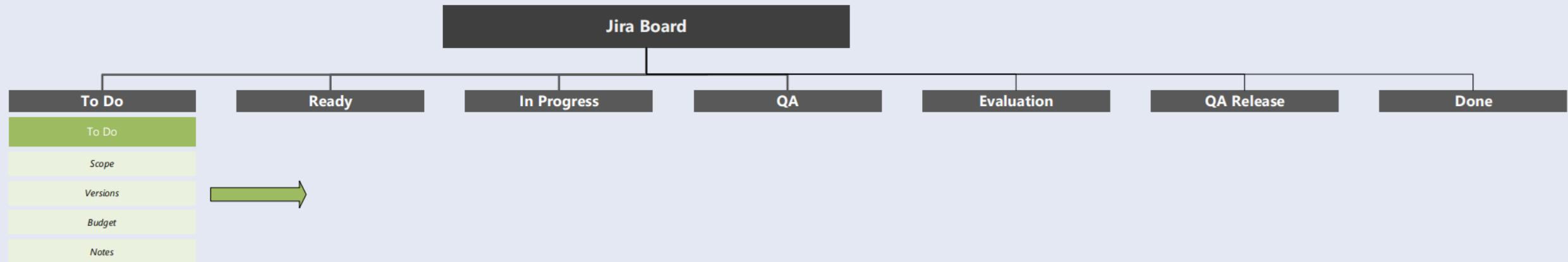
Our repositories

- We maintain compatibility and support across three active versions of our software



Development

3. Jira: To Do → Ready



Development

3. Jira: To Do → Ready

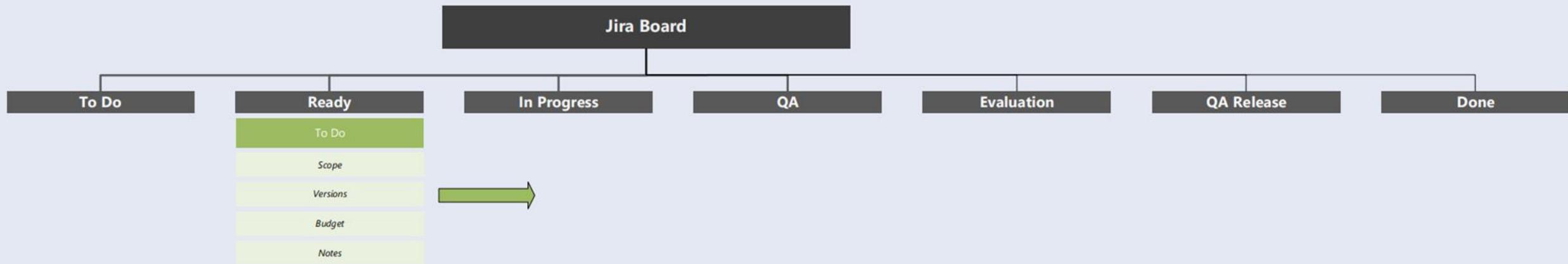
Taking ownership of a work item

- Read the work items description carefully before starting
- Ask clarifying questions to the stakeholder if requirements are unclear
- Proper preparation saves both personal and company time



Development

4. Jira: Ready → In Progress



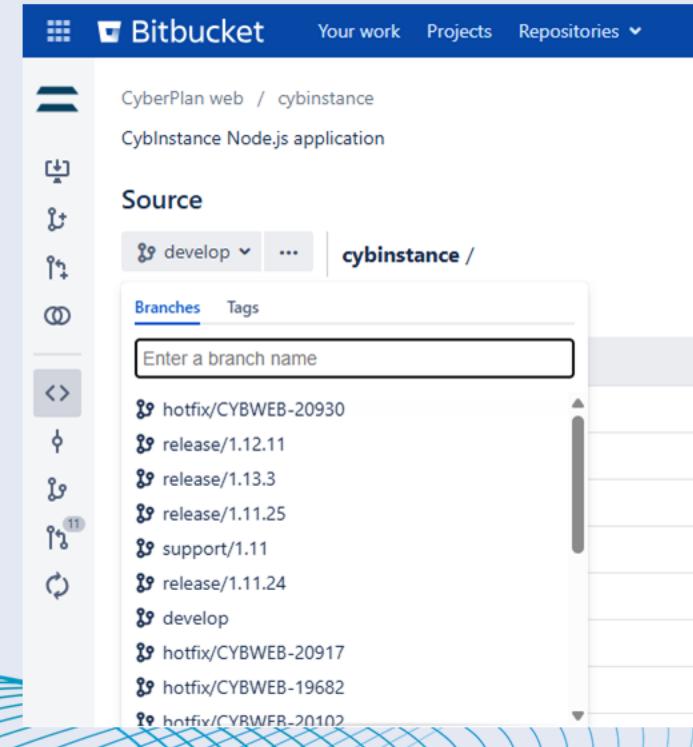
Development

4. Jira: Ready → In Progress

Development Workflow

- When development begins and code needs merging, a new branch is created from one of the three supported versions

```
git checkout -b <work item-id> origin/<version-branch>
```



Development

a. Bitbucket Example

Bitbucket Your work Projects Repositories Search for code, commits or repositories... ? 📈 9 🔍

CyberPlan web / cybinstance

Commits

develop ...

Author	Commit	Message	Commit date	Issues	Builds
Lorenzo Airey	9c897c3b0e9	Pull request #2658: CYBWEB-19278 Reduce critical pt1 Merge in CYBWEB/cybinstance from feature/CYBWEB-19278 to develop * commit '9ebd5e9b6ae4b35b190ba6b'	29 Oct 2025	CYBWEB-19278	
Lorenzo Airey	9ebd5e9b6ae	CYBWEB-19278 QA Fix convert let to const	27 Oct 2025	CYBWEB-19278	
Lorenzo Airey	9c32dc735f6	CYBWEB-19278 QA Fix Linting	24 Oct 2025	CYBWEB-19278	
Lorenzo Airey	62bde881b8b	CYBWEB-19278 Reduce critical pt5	24 Oct 2025	CYBWEB-19278	
Lorenzo Airey	2763a40572e	CYBWEB-19278 Reduce critical pt4	24 Oct 2025	CYBWEB-19278	
Lorenzo Airey	53dade11f5c	CYBWEB-19278 Reduce critical pt3	24 Oct 2025	CYBWEB-19278	
Lorenzo Airey	3607860995d	CYBWEB-19278 Reduce critical pt2	24 Oct 2025	CYBWEB-19278	
Lorenzo Airey	dfdb541d6c5	CYBWEB-19278 Reduce critical pt1	24 Oct 2025	CYBWEB-19278	
	c9b34b4bbfc	M Pull request #2753: Feature/CYBWEB-20563 Merge in CYBWEB/cybinstance from feature/CYBWEB-20563 to develop * commit '1ec7554835f653f6e8c0389eb32ff90428'	27 Oct 2025	7 Jira Issues	

Development

5. Jira: Ready → In Progress

Integrated Quality Checks during Development

- We maintain an active connection to SonarQube, enabling real-time static code analysis
- Developers receive instant feedback on code issues before committing

```
Expected a `for-of` loop instead of a `for` loop with this simple iteration. sonarqubetypescript:S4138
(parameter) header: any
View Problem (Alt+F8) Quick Fix... (Ctrl+.)
    if (header[i].name === alias || header[i].name === `${tableName}.${alias}`) {
        type = header[i];
        break;
```

Development

a. What is SonarQube?

SonarQube: Code Quality and Security Platform

- SonarQube is an open-source platform for continuous inspection of code quality
- Detects bugs, vulnerabilities, and code smells early in the development process
- Works with multiple languages (Java, C#, TypeScript, etc.)



Development

b. SonarQube Example

cybinstance Failed Last analysis: 20 minutes ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines	TypeScript...
14 D	0 A	10 E	0 A	0.0%	2.7%	257 L	

QUALITY GATE STATUS Failed 2 conditions failed

The quality gate used by this project does not comply with Clean as You Code.

Fixing [this quality gate](#) will help you achieve a Clean Code state.

[Learn why](#)

2 condition(s) failed on new code

Reliability Rating on New Code is worse than A

New Major Issues is greater than 0

Lines of Code

Language	Lines of Code
TypeScript	47k
JavaScript	12k
CSS	2k
HTML	1k

Review this redundant assignment: "argidx" already holds the assigned value along all execution paths. 5 years ago ▾ L236 % T redundant

Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment

Refactor this function to reduce its Cognitive Complexity from 29 to the 15 allowed. 5 years ago ▾ L252 % T brain-overload

Code Smell ▾ Critical ▾ Open ▾ Not assigned ▾ 19min effort Comment

Review this redundant assignment: "argidx" already holds the assigned value along all execution paths. 5 years ago ▾ L236 % T redundant

Assignments should not be redundant [typescript:S4165](#)

Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort 0 comments

Where is the issue? Why is this an issue?

The transitive property says that if `a == b` and `b == c`, then `a == c`. In such cases, there's no point in assigning `a` to `c` or vice versa because they're already equivalent.

This rule raises an issue when an assignment is useless because the assigned-to variable already holds the value on all execution paths.

Noncompliant Code Example

```
a = b;
c = a;
b = c; // Noncompliant: c and b are already the same
```

Compliant Solution

Development

6. Jira: Ready → In Progress

Unit tests

- Every new feature or bug fix includes unit tests
- Developers are encouraged to:
 - Cover edge cases
 - Use mocking for dependencies
 - Keep tests isolated and repeatable

Development

a. Jest

Jest for Unit tests

- Jest is a JavaScript testing framework primarily used for unit testing
- Is used extensively in our projects involving JavaScript and TypeScript
- Jest is integrated into the CI/CD pipeline via Bamboo

Why Jest?

- Jest is fast and runs tests in parallel

Development

b. Jest Example

```
You, 42 seconds ago
1 import * as Promise from 'bluebird';
2 import * as Config from 'config';
3 import BackupController from './backupController';
4 import ParameterController from './parameterController';
5
6 describe('setSystemSnapshotCurrentDb', () => {
7
8     const cases = [
9         ['Should return instance.startingDb when instance.database, instance.currentDb, instance.startingDb are defined',
10            { database: 'database.udl', currentDb: 'currentDb.udl', startingDb: 'startingDb.udl' }, 'startingDb.udl'],
11         ['Should return instance.database when instance.database and instance.startingDb are defined, instance.currentDb is not defined',
12            { database: 'database.udl', startingDb: 'startingDb.udl' }, 'database.udl'],
13         ['Should return instance.database when instance.database and instance.currentDb are defined, instance.startingDb is not defined',
14            { database: 'database.udl', currentDb: 'currentDb.udl' }, 'database.udl'],
15         ['Should return instance.database when instance.database is defined, instance.currentDb and instance.startingDb are not defined',
16            { database: 'database.udl' }, 'database.udl']
17     ] as any[];
18
19     test.each(cases)('%s', (scenario, configuration, expectedResult) => {
20
21         jest.spyOn(ParameterController.prototype, 'getConfiguration').mockImplementation(() => Promise.resolve(configuration));
22         jest.spyOn(ParameterController.prototype, 'setParameters').mockImplementation(() => Promise.resolve());
23         jest.spyOn(Config, 'get').mockImplementation(() => Promise.resolve('placeholderInstanceString'));
24
25         return BackupController.setSystemSnapshotCurrentDb()
26             .then((startingDb) => {
27                 expect(startingDb).toBe(expectedResult);
28             });
29     });
30 });
31
32 afterAll(() => {
33     jest.restoreAllMocks();
34 });
35
```

Development

7. Jira: Ready → In Progress

Automated Builds

- After opening a Pull Request, Bamboo triggers an automated build and test pipeline after each commit

The screenshot shows the Bamboo build dashboard for Build #17. The top navigation bar includes 'Bamboo', 'My Bamboo', 'More', 'Create', 'Search', and a help icon. The main area displays a green progress bar indicating the build status. Below the bar, the message '#17 was successful – Scheduled' is shown. A navigation bar at the bottom includes 'Summary', 'Tests', 'Commits', 'Artifacts' (which is underlined), 'Logs', 'Metadata', 'Webhooks', and 'Issues'. The 'Artifacts' section heading is 'Artifacts', with the sub-section 'Job artifacts' below it. A detailed description of artifacts is provided, stating: 'An artifact is something created by a job build (e.g. Jar files). Artifact definitions are used to specify which artifacts to keep from a build and are configured for individual jobs.' A table lists the produced artifacts: 'Default Job Default Stage' with 'archives' as the artifact type and a file size of '473 MB'.

Produced in job	Artifact	File size
Default Job Default Stage	archives	473 MB

The screenshot shows a Jira issue titled 'bugfix/CYBWEB-11029 → release/1.9.0' with an 'OPEN' status. A modal window titled 'Builds' is displayed, showing a single successful build entry: 'CyberPlan web - CybInstance - bugfix-CYBWEB-11029' completed 7 minutes ago with 28 tests in 16 minutes. The 'Close' button is visible in the bottom right corner of the modal.

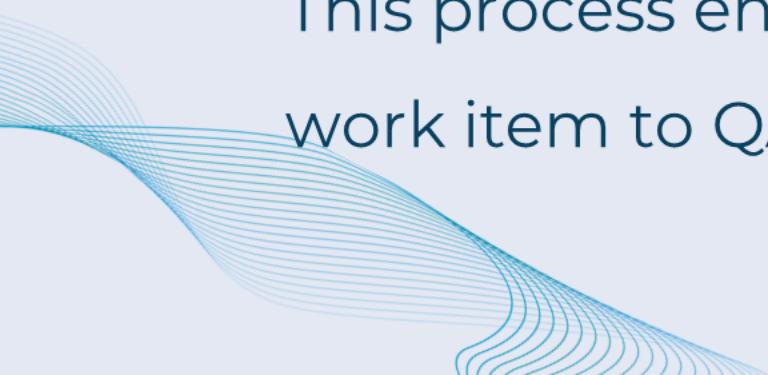
Development

a. Bamboo: Build Results

Results are then sent back to the developer:

- ✓Pass: Code is ready for merge
- ✗Fail: Developer receives detailed feedback on:
 - Unit test failures
 - Linting or code style issues
 - Security vulnerabilities

This process ensures developers fix issues before sending the work item to QA, improving stability and quality



Development

b. Bamboo: Build Failure

```

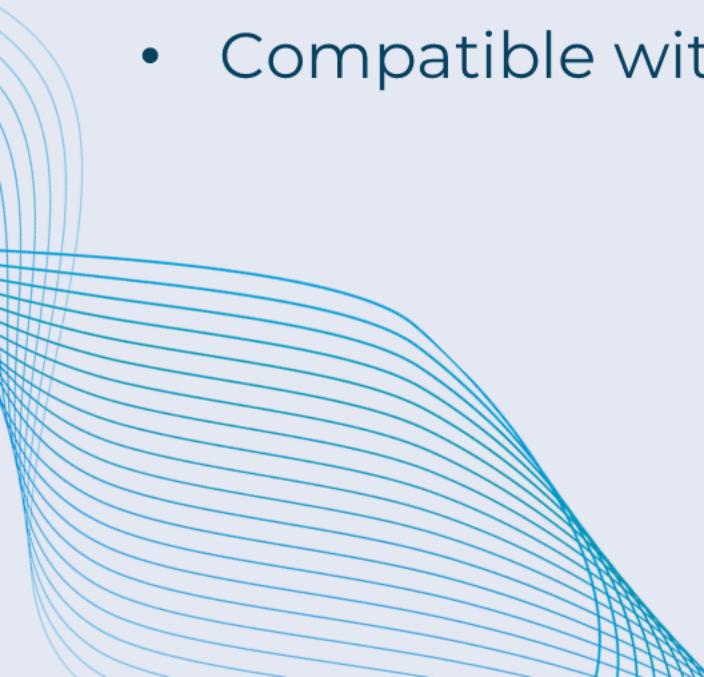
at com.atlassian.bamboo.security.ImpersonationHelper.runWithSystemAuthority(ImpersonationHelper.java:17)
at com.atlassian.bamboo.security.ImpersonationHelper$1.run(ImpersonationHelper.java:41)
at java.lang.Thread.run(Thread.java:748)
Caused by: com.atlassian.bamboo.plugins.stash.repository.StashRepositoryException
at com.atlassian.bamboo.plugins.stash.v2.BitbucketServerWorkingCopyManager.retrieveSourceCode(BitbucketServerWorkingCopyManager.java:100)
at com.atlassian.bamboo.plugins.vcs.task.VcsCheckoutTask.access$000(VcsCheckoutTask.java:54)
at com.atlassian.bamboo.plugins.vcs.task.VcsCheckoutTask.fillWorkingDirFromVcs(VcsCheckoutTask.java:206)
at com.atlassian.bamboo.plugins.vcs.task.VcsCheckoutTask$1.call(VcsCheckoutTask.java:126)
at com.atlassian.bamboo.plugins.vcs.task.VcsCheckoutTask$1.call(VcsCheckoutTask.java:123)
at com.atlassian.bamboo.executor.RetryingTaskExecutor.retry(RetryingTaskExecutor.java:112)
...
Caused by: com.atlassian.bamboo.plugins.git.GitCommandException: command ['C:\Program Files\Git\cmd\git.exe' log -1 --encoding utf-8] in directory C:\Windows\system32 failed: fatal: bad object 73a8615581f402cb1860198bc4f217992ac189e0
at com.atlassian.bamboo.plugins.git.GitCommandProcessor.runCommand(GitCommandProcessor.java:740)
at com.atlassian.bamboo.plugins.git.GitCommandProcessor.runCommand(GitCommandProcessor.java:669)
at com.atlassian.bamboo.plugins.git.GitCommandProcessor.getShaOf(GitCommandProcessor.java:432)
at com.atlassian.bamboo.plugins.git.GitCommandProcessor.getShaOf(GitCommandProcessor.java:404)
at com.atlassian.bamboo.plugins.git.NativeGitOperationHelper.checkRevisionExistsInCacheRepository(NativeGitOperationHelper.java:100)
at com.atlassian.bamboo.plugins.git.v2.GitWorkingCopyManager.lambda$checkout$1(GitWorkingCopyManager.java:238)
at io.atlassian.util.concurrent.ManagedLocks$ManagedLockImpl.withLock(ManagedLocks.java:233)
at com.atlassian.bamboo.plugins.git.v2.GitWorkingCopyManager.checkout(GitWorkingCopyManager.java:235)
at com.atlassian.bamboo.plugins.git.v2.GitWorkingCopyManager.retrieveSourceCode(GitWorkingCopyManager.java:122)
...
Failed to prepare the build 'CyberPlan web - CybInstance - bugfix-CYBWEB-11010 - Default Job #17 (CYBWEB-CYBINS80-JOB1-17)'

```

Pipelines (CI/CD)

1. What is Bamboo?

- Bamboo is a CI/CD (Continuous Integration / Continuous Delivery) tool developed by Atlassian
- Automates build, test, and deployment processes
- Works seamlessly with Jira, Bitbucket, and other Atlassian tools
- Compatible with multiple languages and frameworks



Pipelines (CI/CD)

2. Bamboo: Continuous Integration and Delivery Tool

- Reduces human error in releases
- Speeds up delivery cycles
- Ensures consistent quality across environments



Pipelines (CI/CD)

3. Bamboo: Continuous Integration and Delivery Tool

- After opening a Pull Request, Bamboo triggers an automated build and test pipeline after each commit
- Additionally, every night, if new code has been merged, Bamboo triggers a nightly build
 - This build validate the latest development state
 - Ensure all tests and checks pass before the next workday
 - If the build succeeds, the most up-to-date version is ready for developers at the start of the day

Pipelines (CI/CD)

4. Bamboo: Nightly Build Example

Bamboo My Bamboo Projects Build Deploy More Create Search

Build dashboard / CyberPlan web / CyberPlan web 1.9 (release)

Build #39

Plan branch: release/1.9.0

#39 was successful

Summary Tests Commits Artifacts Logs Metadata Webhooks Issues

Build result summary

Details

Completed

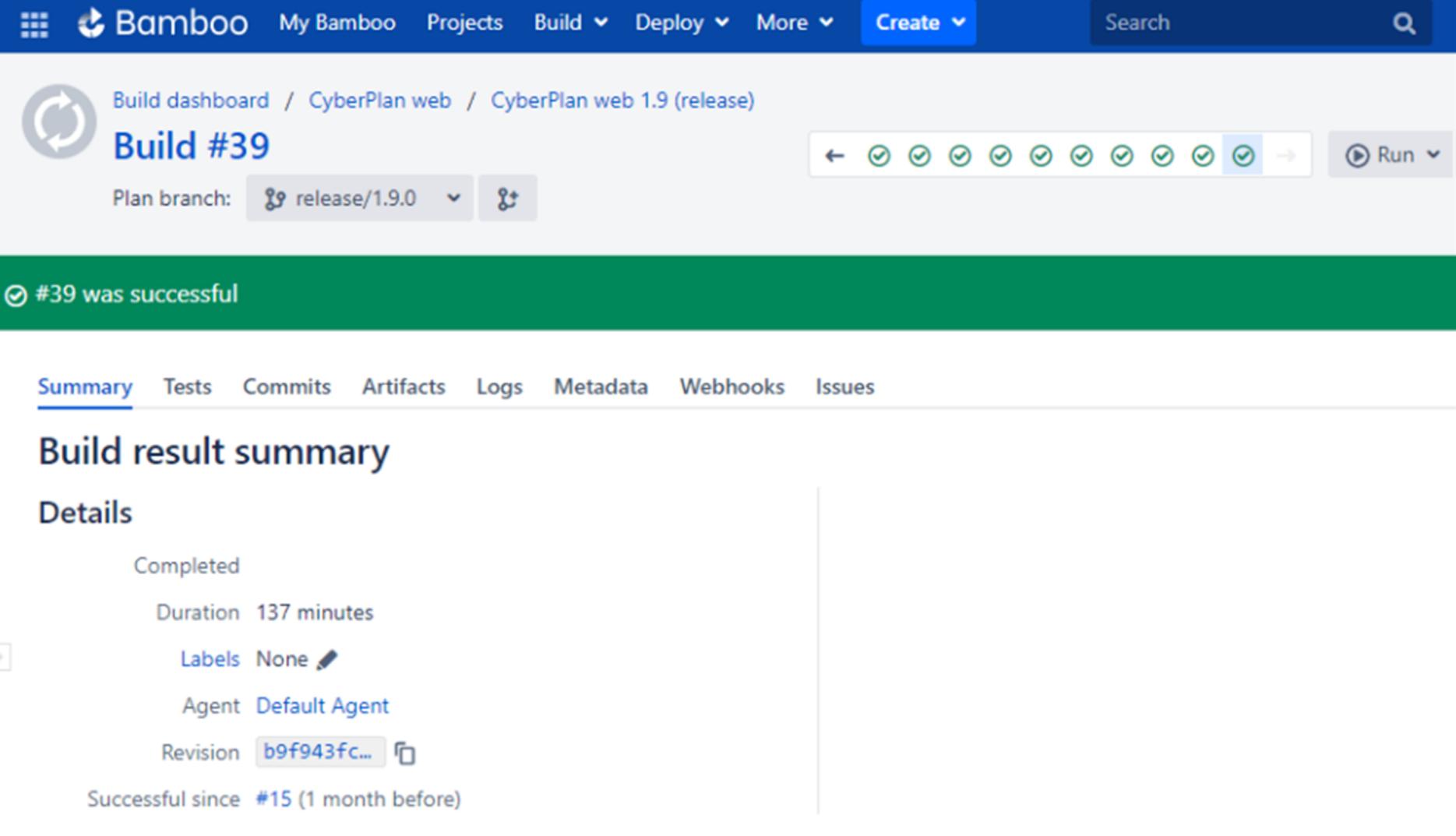
Duration 137 minutes

Labels None

Agent Default Agent

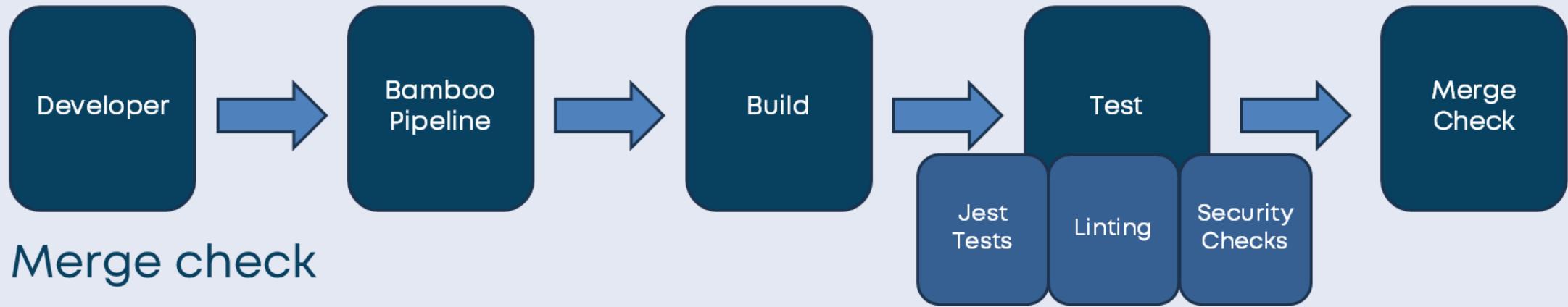
Revision b9f943fc...

Successful since #15 (1 month before)



Pipelines (CI/CD)

5. Bamboo: Workflow



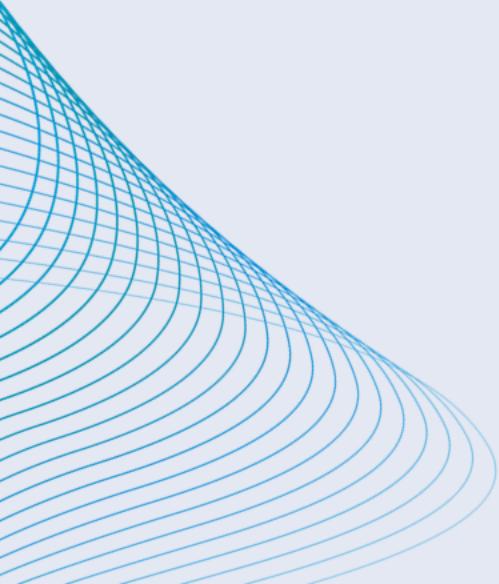
Merge check

- Merge is blocked until the build is successful
- This ensures:
 - Only code that passes all automated checks (build, tests, linting, security) can be merged
 - Prevents broken or unstable code from entering the related support/release/develop branch

Pipelines (CI/CD)

6. Bamboo: How does it work?

- Each version (support/release/develop) has multiple Repositories
- Every Repository has its own Pipeline



Pipelines (CI/CD)

7. Bamboo: How does it work?

Inside each Repository

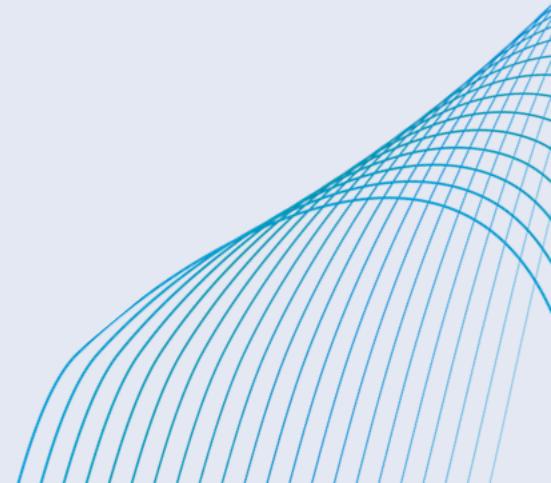
1. Bamboo checks out a specific branch
2. Starts a Docker image of the Branch
3. Clone the source files inside a directory not shared through host and Container (Isolate the build environment)
4. Download node modules
5. Runs unit tests
6. Executes gulp dist (to bundle the output for distribution)

Pipelines (CI/CD)

8. Bamboo: How does it work?

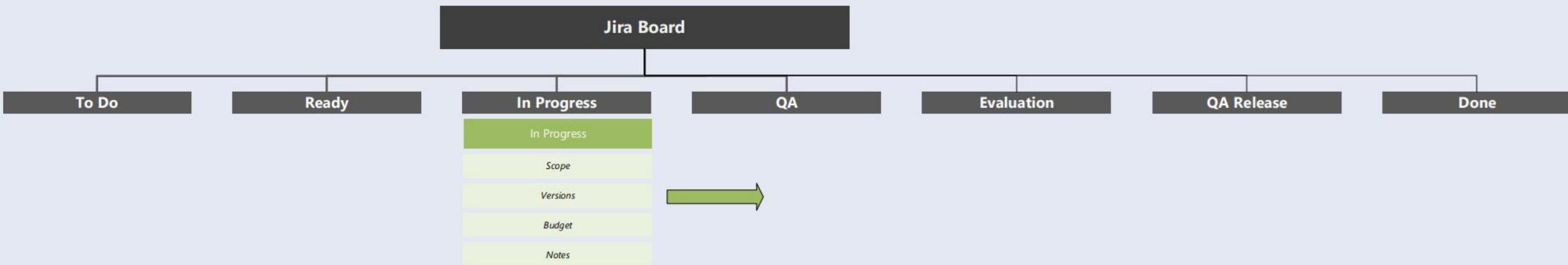
Inside each Repository

7. Clone outputs inside a shared directory through host and container
8. Parses test results to determine if the tests has been successful
9. Final build artifacts are stored in Bamboo so they can be downloaded by developers or delivered



QA (Quality Assurance)

1. Jira: In Progress → QA



QA (Quality Assurance)

1. Jira: In Progress → QA

Validate the software in a controlled environment before merging

- **Tester Role**

QA is done by another developer (tester), who is an expert in the domain related to the work item

- **Local Environment Validation**

Tester runs the code on their local environment to replicate real conditions

QA (Quality Assurance)

2. Jira: In Progress → QA

Key QA Activities

1. Code Review

- Tester reviews the Pull Request (PR)
- Adds comments or tasks if improvements are needed

2. Functional Verification

- Ensure all new features work as expected
- Validate workflows and user scenarios

3. Regression Check

- Confirm that new changes haven't broken existing functionality

QA (Quality Assurance)

3. Jira: In Progress → QA

Key QA Activities

4. Run Unit Tests

- Validate new code and existing functionality
- Ensure coverage for edge cases

5. Local Build Verification

- Confirm no local build errors

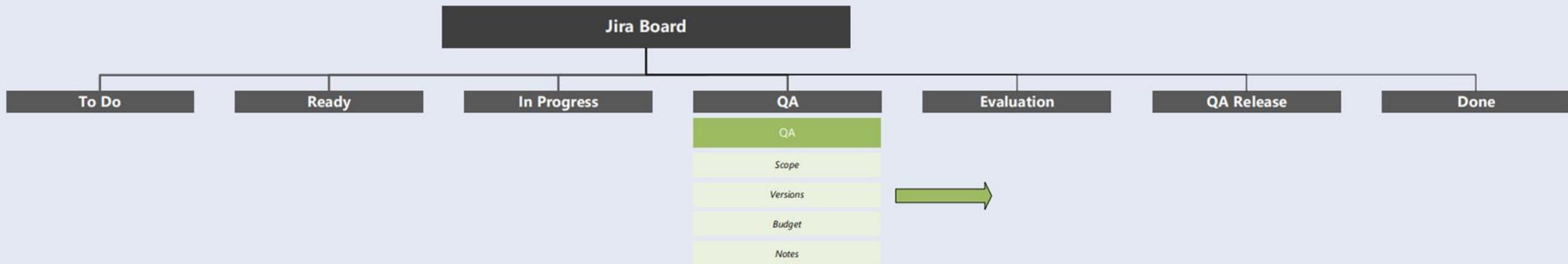
6. Bamboo Build Verification

- Confirm no automated build errors

7. Approves the PR

Evaluation

1. Jira: QA → Evaluation



Evaluation

1. Jira: QA → Evaluation

Ensure code quality and functional correctness before merging into main branches

- Performed after developer QA and before QA Release
- Conducted by the Team Lead of the project
- Ensures compliance with coding standards and functional requirements

Evaluation

2. Jira: QA → Evaluation

Key Evaluation Activities

1. Code Review

- Team Lead reviews the Pull Request
- Adds comments or tasks for improvements

2. Functional Verification

- Ensure all new features work as expected
- Validate workflows and user scenarios

3. Run Unit Tests

- Ensure coverage for edge cases

Evaluation

3. Jira: QA → Evaluation

Key Evaluation Activities

4. Bamboo Build Verification

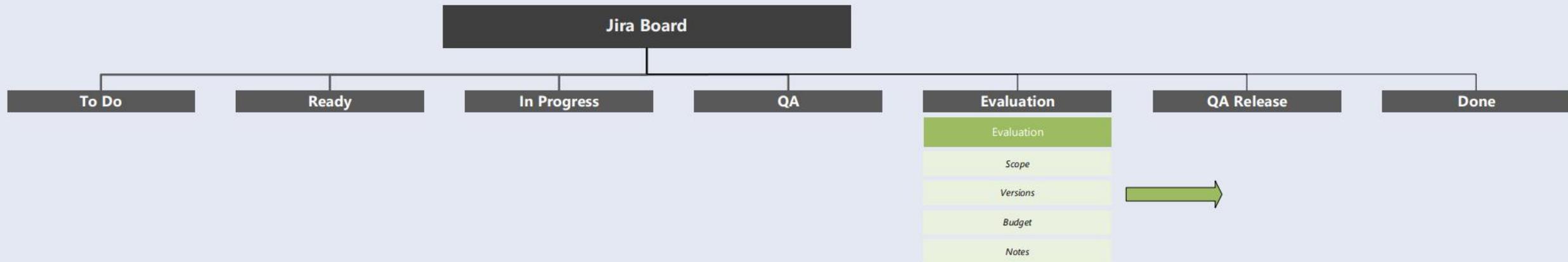
- Confirm no automated build errors

5. Approval

- If everything passes → Merge allowed
 - Approval check on PR 
 - If issues found → Sent in Development again

QA Release

1. Jira: Evaluation → QA Release



QA Release

1. Jira: Evaluation → QA Release

Validate the product in an environment that closely mirrors production before final release

Who Performs QA Release?

- Product Specialists

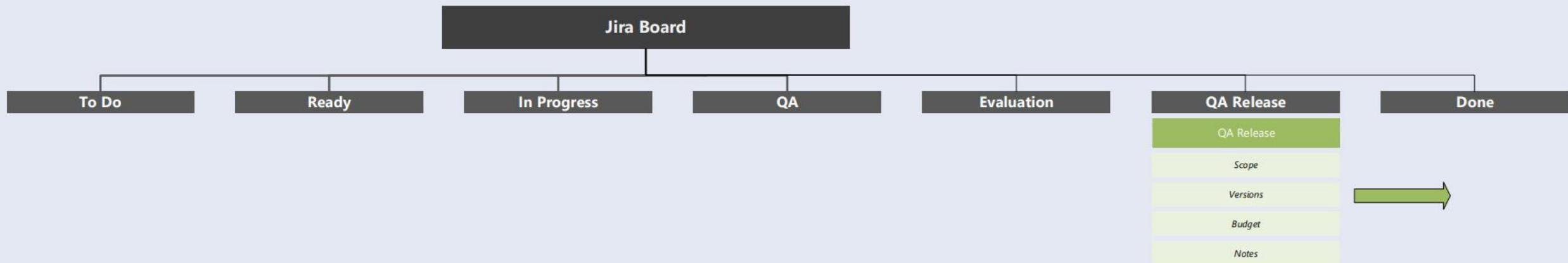
Experts in the business domain and real-world workflows

- Similar Production Environment

- Same configurations as production
- Realistic data sets
- Controlled conditions for accurate validation

QA Release

2. Jira: QA Release → Done



Release

Release Process Overview

- Structured Workflow

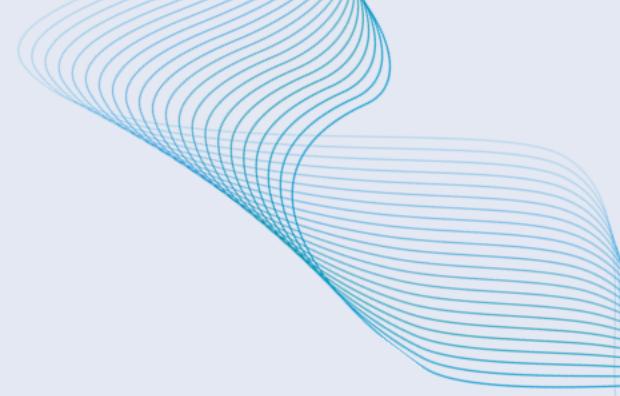
To deliver high-quality and secure releases

- Risk minimization

The process reduces risks and human errors to ensure reliability during the software release



Release



Release Steps

1. Unit Testing

Test Boost: C++, Test Jest: Typescript

2. Integration testing

Interaction between software modules to simulate Real-World Scenarios

3. Security Analysis

Dependencies scan with Snyk for vulnerabilities

Release

Release Steps

4. Build Creation

5. Installation and Update Testing

To ensure smooth installation and compatibility with previous software versions

6. Publish Documentation

Technical Documentation, User Manual, Patch Notes

7. Release Distribution



Lesson Learned as a Software Engineer

1. My experience

- Critical Thinking
 - Ability to analyze problems deeply and question assumptions
- Reading Code and Documentation
 - Developing strong skills to understand existing codebases and technical docs
- Avoid Unverified Assumptions
 - Always validate before acting to prevent errors and rework

Lesson Learned as a Software Engineer

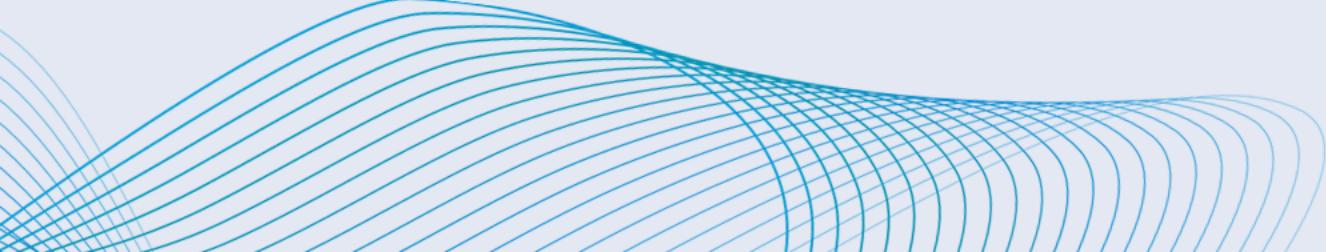
2. My experience

- Managing Constraints and Balancing Cost/Benefit

Making decisions that optimize resources while meeting requirements

- Continuous Improvement Mindset

Embracing learning and iterative enhancement as a core principle



Thanks for your attention

CYBERTEC
—
ZUCCHETTI

 Opera MES
CYBERPLAN

Telefono: +39 040 898111
Email: hr@cybertec.it
Web: www.cyberplan.it