eficode

# Testing in DevOps - how to assess teams and give guidance

Szilard Szell, 5G DevOps transformation lead

# Szilárd Széll – intro


Responsibilities

- 5G DevOps transformation lead and Senior Consultant at Eficode
- Test Coach and DevOps Evangelist in NOKIA (19 years experience)
- Former President of the Hungarian Testing Board
- Processes Management and Compliance Work Group Chair of ISTQB
- Program Committee Chair and Member of – UCAAT and HUSTEF

Certifications

- DevOps DASA Foundation, SAFe SPC, Certified Scrum Master
- ISTQB CTEL-ITP-Full, CTAL-TM, CTFL-AT, CTFL
- IREB CPRE
- Lean Six Sigma Green Belt
- Lean Service Creation - Facilitator

eficode

## Index


# DEVOPS IN A NUTSHELL

eficode.com

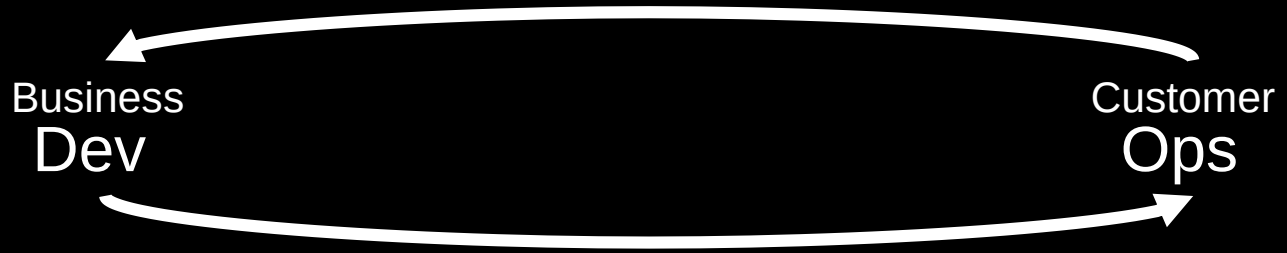
# 1. The First Way: The Principles of Flow

# 1. The First Way: The Principles of Flow

“Being able to take needless work out of the system is more important than being able to put more work into the system.”

Gene Kim

# 2. The Second Way: The Principles of Feedback

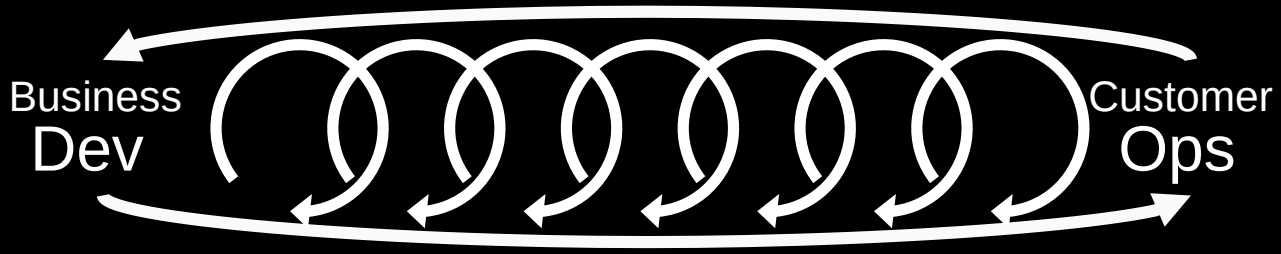


# 2. The Second Way: The Principles of Feedback

“Improving daily work is even more important than doing daily work.”

Gene Kim

# 3. The Third Way: The Principles of Continual Learning and Experimentation

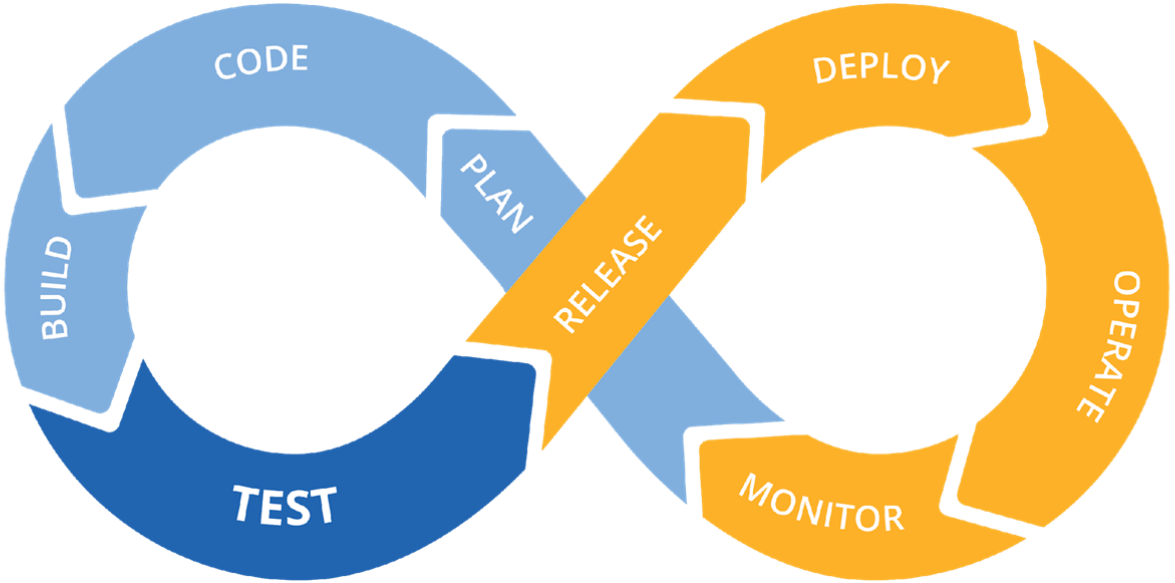


# 3. The Third Way: The Principles of Continual Learning and Experimentation

“If you can’t out-experiment and beat your competitors in time to market and agility, you are sunk.”

Gene Kim

# one DevOps view




- Image source: DevOps | freeCodeCamp Guide

eficode

# Eficode's DevOps view

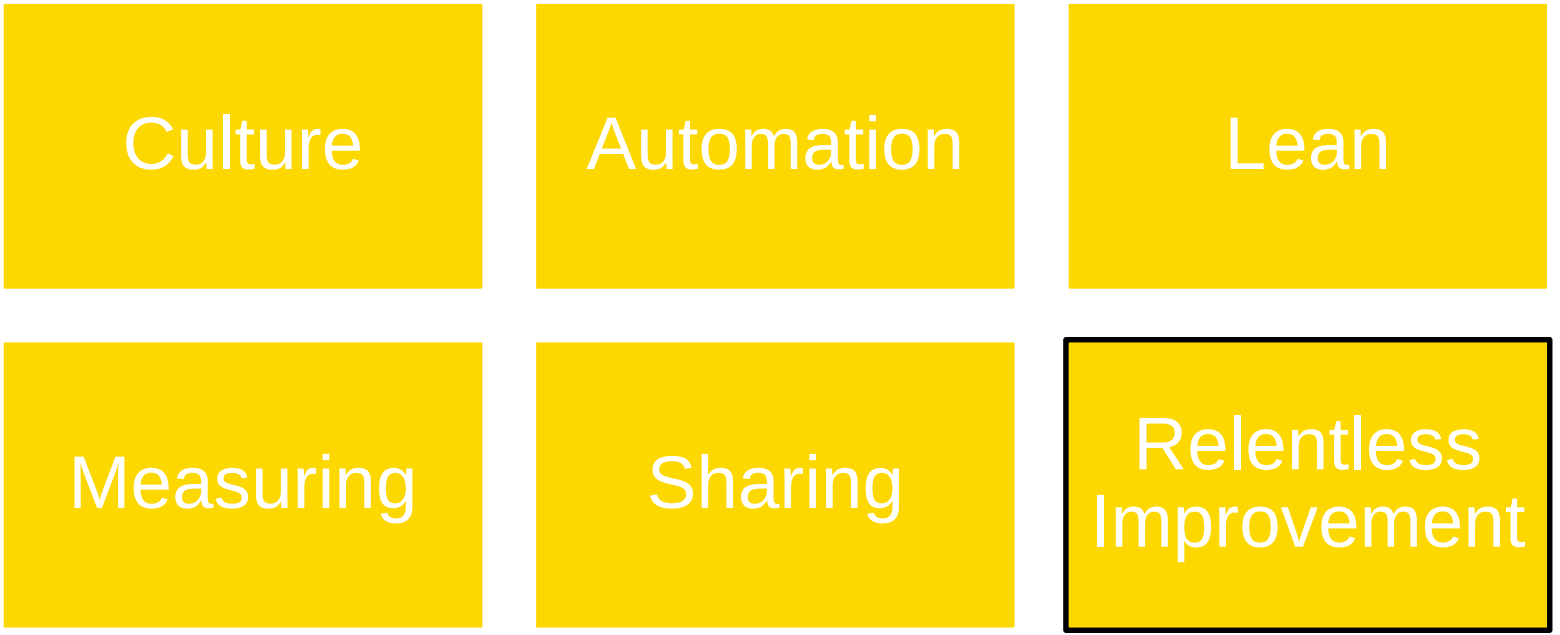
Using technology to make organization's operations more agile and reactive to customer needs

eficode

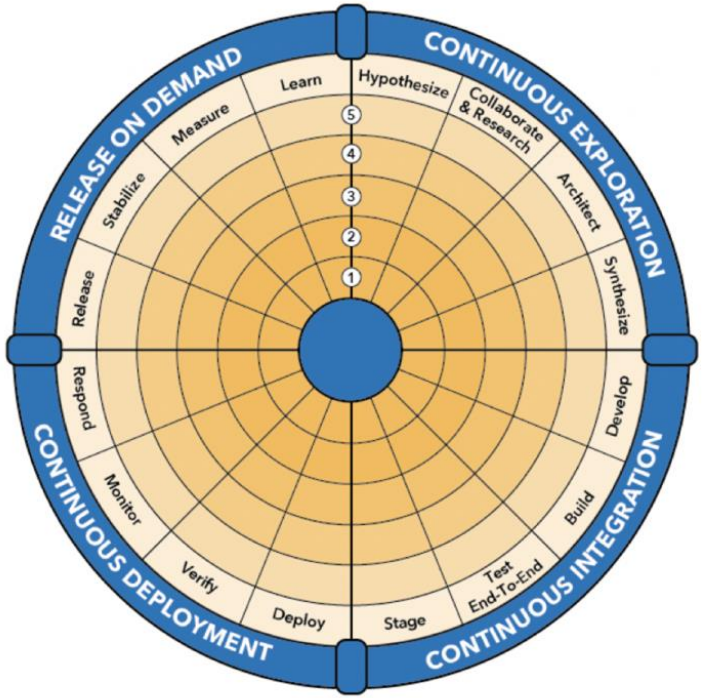

# DEVOPS ASSESSMENT(s)

eficode.com

# DevOps – CALMS or CALMR model



# SAFe DevOps and Release on Demand Health Radar

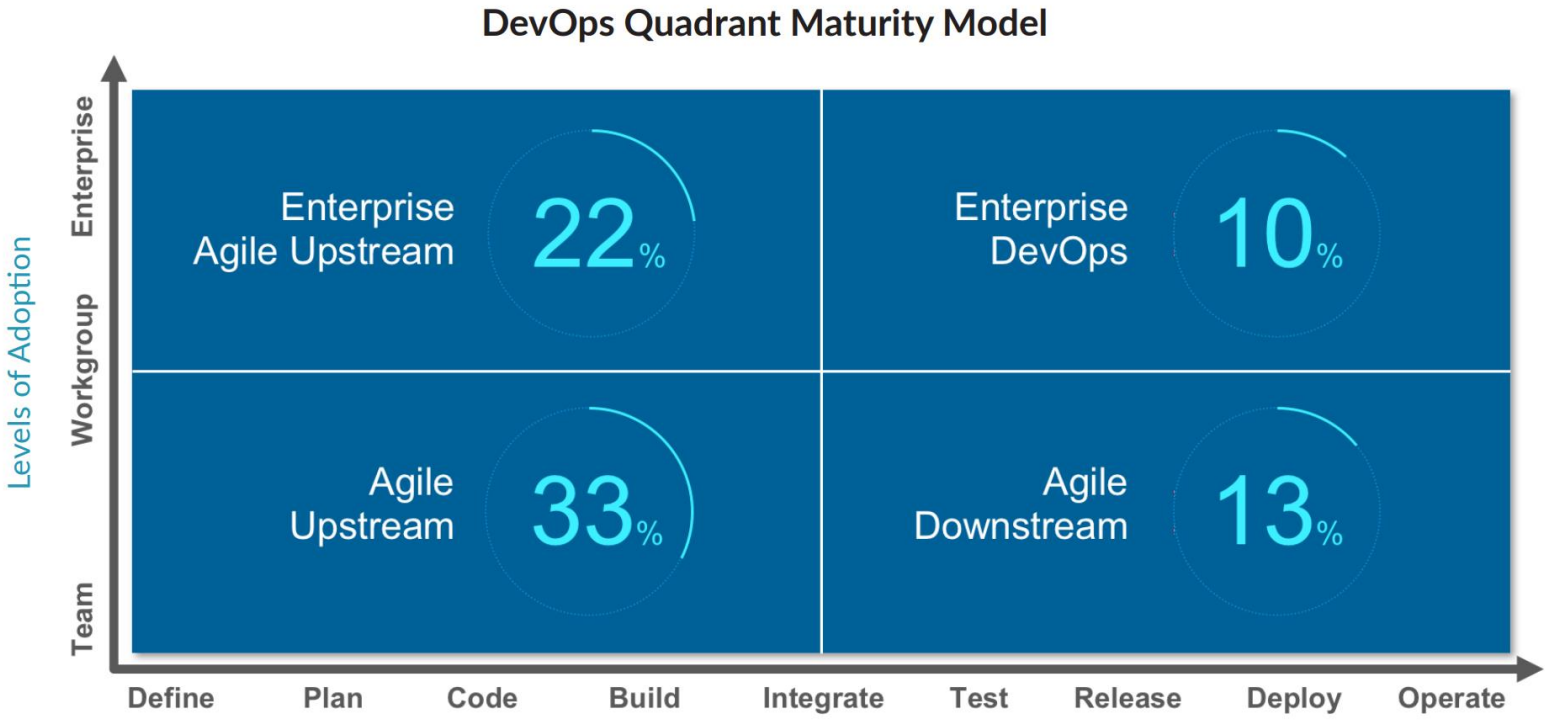





- Continuous Exploration
- Continuous Integration
- Continuous Deployment
- Release on Demand

eficode

# CouldBees DevOps Quadrant




eficode

# DEVOPS

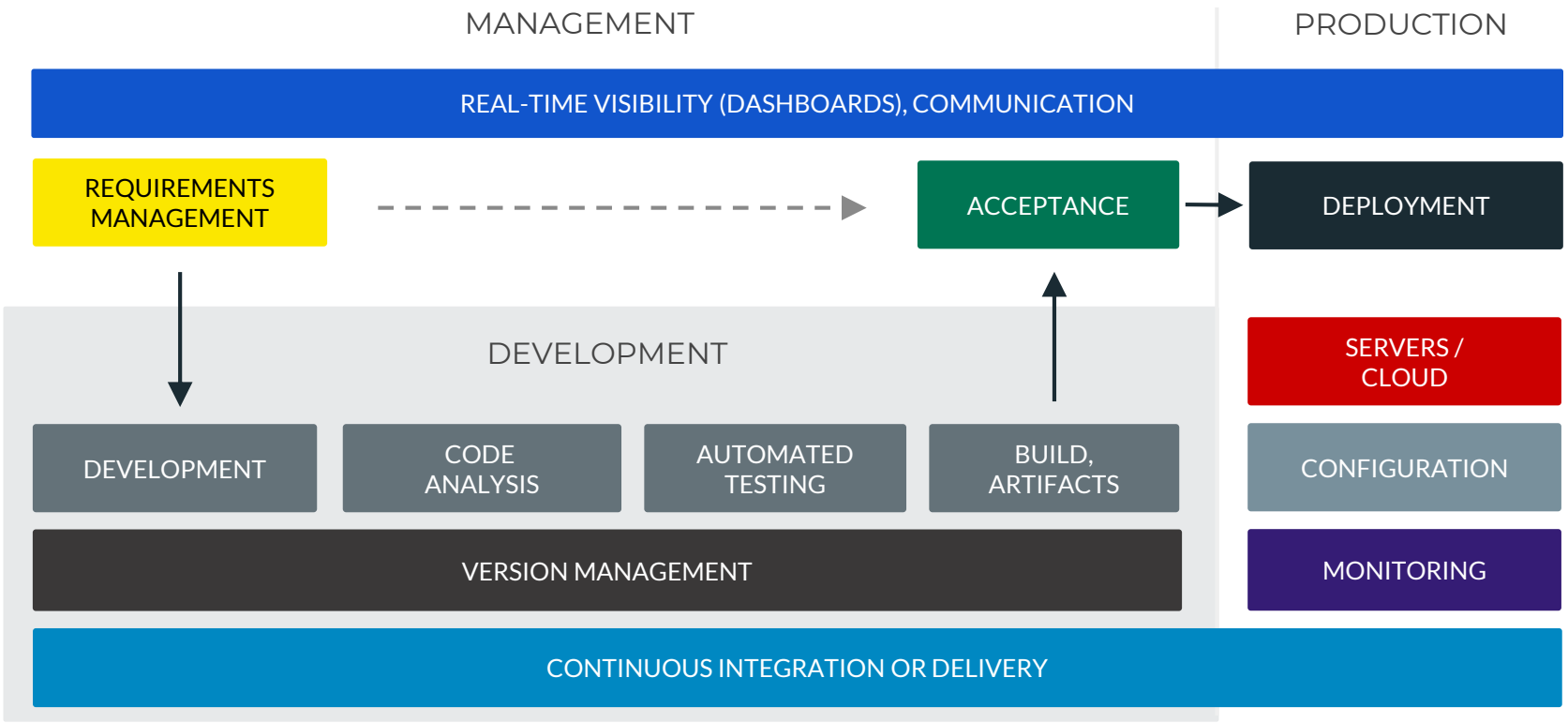
## eficode Maturity model

**DEVOPS MATURITY 68 / 100**

| | 001 | 002 | 003 | 004 |
|---|---|---|---|---|
| ENABLEMENT | Development operations have been separated from the business knowledge. Starting a new development project is laborious. | Starting new development projects is agile, and there are practices in place for steering the project. | New projects can be connected to organization's strategic targets. Starting a new pilot project is easy. | Real-time metrics supporting decision making and tracking the completion of strategic targets are available. |
| ORGANIZATION AND CULTURE | Design, development and quality assurance are separate from each other. Communication is primarily in writing. | Work is conducted in teams but development and quality assurance are separate from each other. | The teams work independently. They have total liability for the development and quality assurance of features. | The teams communicate with each other regularly and work together to improve their practices. Communication with the IT operations is continuous. |
| ENVIRONMENTS AND RELEASE | Products are environment-specific and they are compiled manually. Environments are installed and configured manually. | The system is divided into parts and the compiling environment is known. Some releases are automated. | Environments can be installed and configured automatically. Build and release processes are automated. | Releases may be conducted automatically and continuously. Migration and recovery processes work as expected. |
| BUILDS AND CONTINUOUS INTEGRATION | Product integration is automatic, but configuration and deployment are controlled manually. No artifact or change logs management. | The product is built by team specifically after every change. Tools are shared. Integration does not involve testing. | Integration covers the entire product and it is connected to acceptance testing. Dependencies are known and managed. | Build and integration processes are continuously improved based on collected metrics with aim to speed up the feedback cycle and improve visibility. |
| QUALITY ASSURANCE | Quality assurance is conducted completely by hand and primarily after development. | Unit testing or static code analysis is in place for some parts of the product. | Features visible to the end users are covered with automatic tests. Testers participate in the development process. | Acceptance tests present system requirements clearly and guide the development of the system as much as possible. |
| VISIBILITY AND REPORTING | Reports are made by hand when necessary. | Code integration, unit testing and code analysis are visible to the team. | The status of requirements can be monitored in real time in relation to tests and released features. | Real-time metrics are automatically collected from the product development process and used as a basis for improvement. |
| TECHNOLOGIES AND ARCHITECTURE | Technologies and tools are obsolete or are not fit for current requirements. | Technologies are growing old and the architecture is only partially adaptive or the interfaces are lacking. | Technologies are modern or well supported. The interfaces are well documented and exist for all key functionalities. | The architecture and technologies are optimal and enable reaching business targets efficiently. |

EXAMPLE

# eficode DevOps maturity assessment scope




eficode


# TESTING ASPECT OF DEVOPS

eficode.com

# What is important for QA in DevOps?

- Value driven testing built on customer collaboration

Test First Thinking

Continuous Product level Quality Assurance

Shift Left

Automation of Testing tasks

Share (testing) tools and assets across the Value Stream

eficode

# What is important for QA in DevOps?

- Value driven testing built on customer collaboration

Test First Thinking

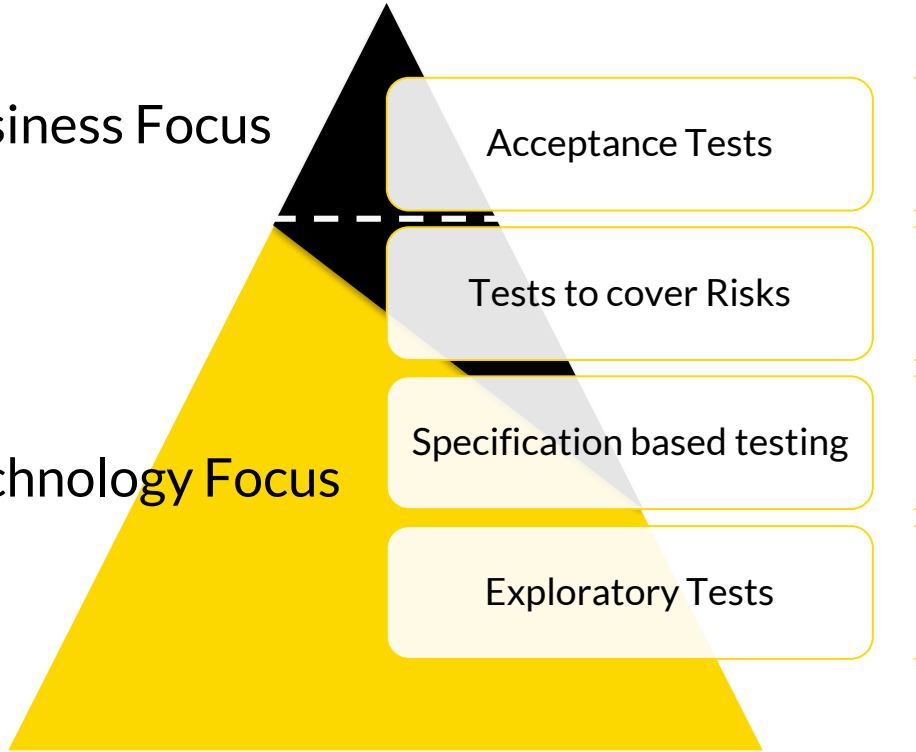
- Feature level acceptance criteria are defined upfront collaboratively
- All types of test are identified upfront collaboratively
- Testing is guided by Risk

eficode

# Collaborative Test Design

Production Acceptance Test

In house testing




**Acceptance Tests**

- ATDD or BDD, resulting in Automatic Acceptance Tests
- Focusing on Functional aspect

**Tests to cover Risks**

- Failure Mode and Effect Analysis to capture negative scenarios
- Risk Based Testing to focus effort

**Specification based testing**

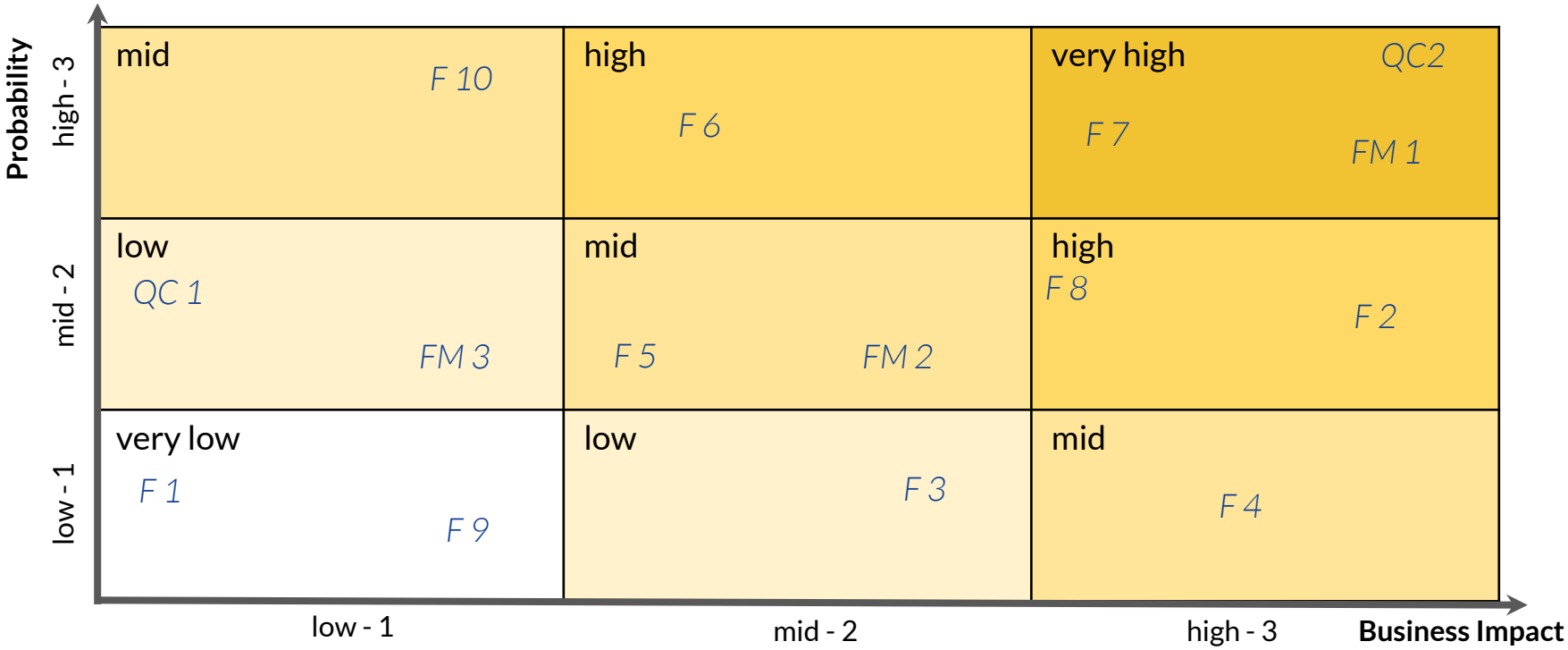
- Requirement Based Test Design on all level of Requirements, with traceability
- Focusing on Non-functional aspect as well

**Exploratory Tests**

- Experience Based test design focusing on Rainy Day scenarios
- Continuous learning of our system

eficode

# Testing is guided by Risk Levels




*Fx – Feature x or Functional Requirement*
*QCx – Quality Characteristic or Non-functional Requirement*
*FMx – Failure mode*

eficode

# Differentiated Test Approach for different Risk levels – example by Rex Black

| Aggregate Risk | Extent of Testing | Comments |
|---|---|---|
| Very low | None | Only report bugs observed in these risk areas. |
| Low | Opportunistic | Leverage other tests or activities to run a test or two of an interesting condition in the related risk area, but only if it involves a very small investment of time and effort and only if the opportunity presents itself. |
| Medium | Cursory | Run a small number of tests that sample the most interesting conditions in the related risk areas. |
| High | Broad | Run a medium number of tests that exercise many different interesting conditions in the related risk areas. |
| Very high | Extensive | Run a large number of tests that are both broad and deep, where deep tests exercise many combinations and variations of interesting conditions. |

# What is important for QA in DevOps?

- Value driven testing built on customer collaboration

Continuous Product level Quality Assurance

- Coverage and Traceability
- Non-functional QA
- Regression set is updated regularly

eficode

# What is important for QA in DevOps?

- Value driven testing built on customer collaboration

Shift Left

- Static Code Analysis is in use
- Module/Unit testing is in use
- Collaborative, cross-functional Test Planning

# What is important for QA in DevOps?

- Value driven testing built on customer collaboration

Automation of Testing tasks

- Test case as Software
- Automated Test results analysis and reporting
- Test environment deployment automation

eficode

# What is important for QA in DevOps?

- Value driven testing built on customer collaboration

Share (testing) tools and assets across the Value Stream

- Reuse of Test Assets
- Standardized pipeline and tools in use


# TAKE AWAY

eficode.com

## Learning points

1. Big size transformation needs a dedicated team and support
2. There are many DevOps assessments available, choose the one for you
3. Testing and CI/CD is key in DevOps, but there are more key areas
4. Continuous Exploration and Learning is a must

eficode

# Thank You

For more information,
please contact

**Szilard Szell**
szilard.szell@eficode.com