REPUBLIC OF TÜRKİYE
MINISTRY OF INDUSTRY
AND TECHNOLOGY

#NATIONAL TECHNOLOGY INITIATIVE

TÜBİTAK
BİLGEM

15 YEARS
AT THE HEART OF
TECHNOLOGY

TÜBİTAK BİLGEM

Data-Driven Insights for Software Development Process Improvement: A Defect Analysis

Melike TAKIL, Zeliha DİNDAŞ

E-mail : melike.takil@tubitak.gov.tr , zeliha.dindas@tubitak.gov.tr

Article Number : 10034

IARIA

Who are we?

Melike TAKIL

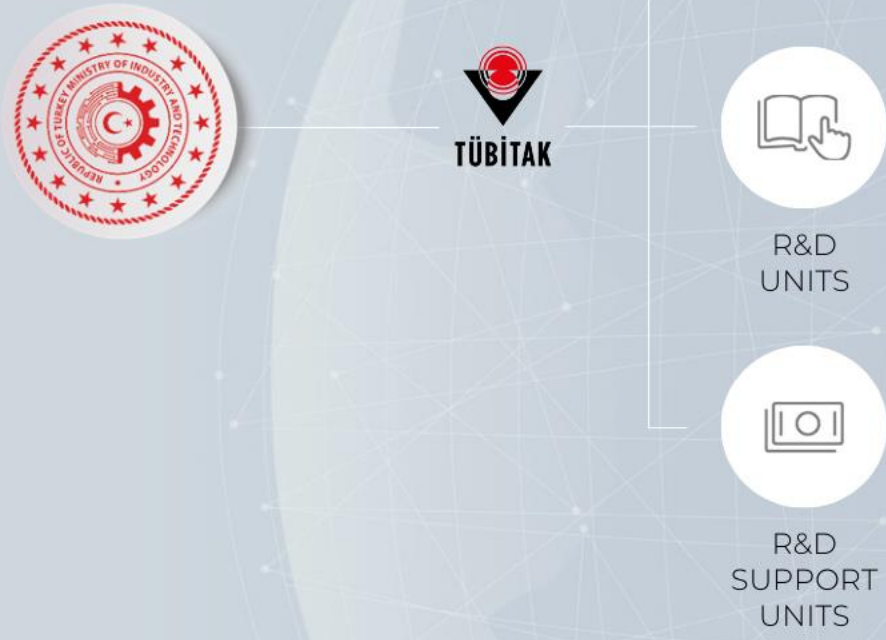
- MSc in Industrial Engineer with 5 years of experience in software industry & public institutions
- Master Thesis: Self-Starting Control Charts for Software Development Projects, 2023.

Zeliha DİNDAŞ

- MSc in Statistics with 10+ years of experience in the public sector.
- Master Thesis : Estimation of Variance Components in Measurement Systems Analysis, 2017.

TÜBİTAK **BİLGEM**

The largest research center in Türkiye,
focusing on **informatics and information security.**



TÜBİTAK BİLGEM

BİLGEM
Informaticsand Information Security Research Center

BİLGEM in Numbers

2000+
Employees

14%
Ph.D
Degree

25%
Master's
Degree

6
Institutes

8
Test Laboratories

200+
Products

230
Projects

+1 Billion $
Portfolio

15
Areas of
Critical Research

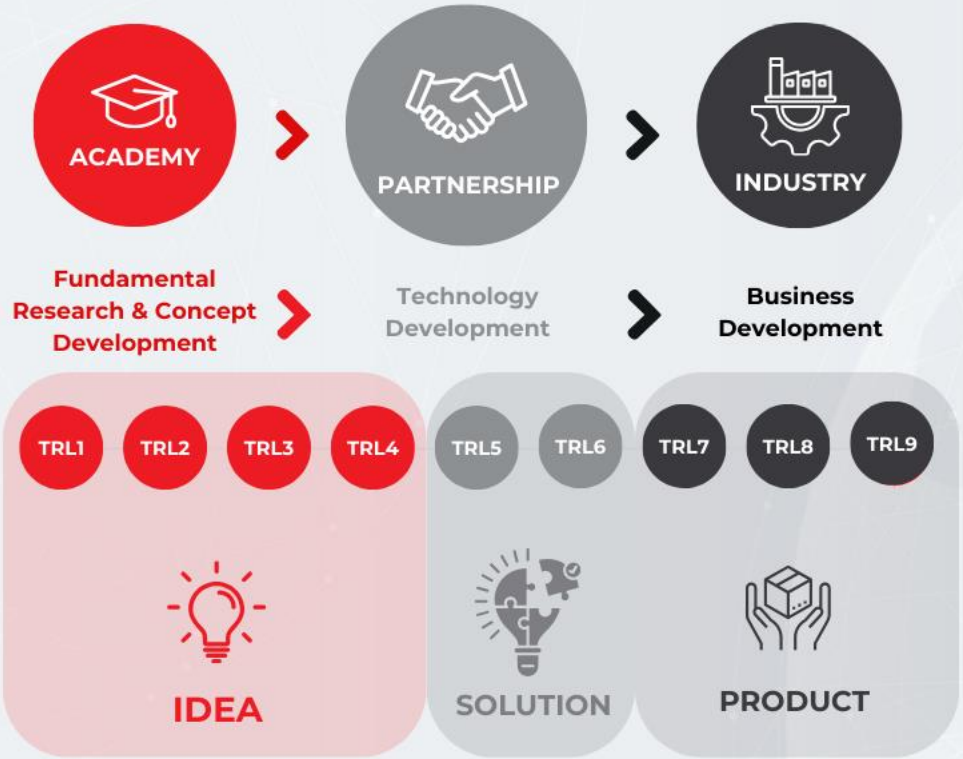
Enabling Technologies and **Research Areas**

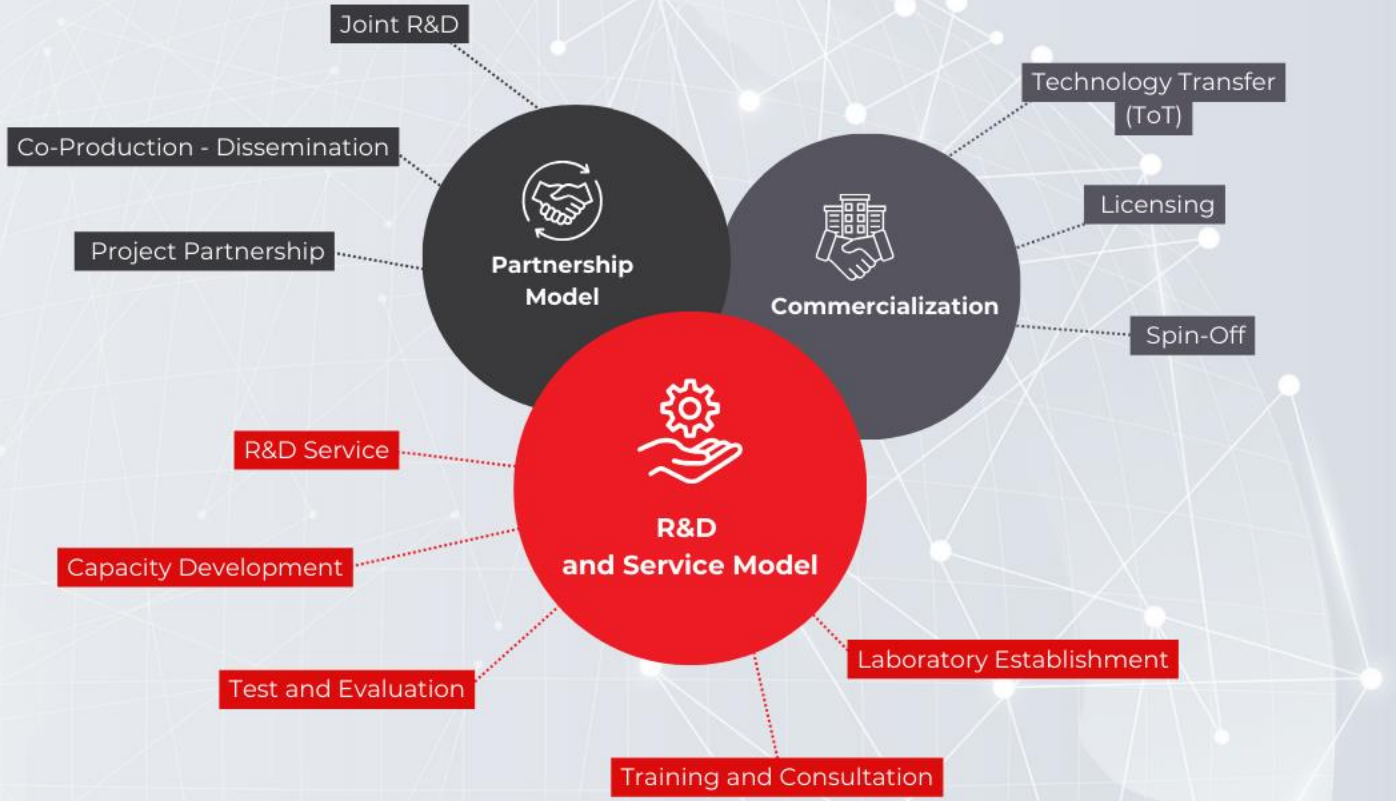
- **ARTIFICIAL INTELLIGENCE**
- **COMMUNICATION TECHNOLOGIES**
- **CYBER SECURITY**
- **CLOUD COMPUTING TECHNOLOGIES**
- **QUANTUM TECHNOLOGIES**
- **SEMICONDUCTOR TECHNOLOGIES**
- **OPERATING SYSTEMS**
- **BLOCKCHAIN TECHNOLOGIES**

- ELECTRONIC WARFARE
- DIGITAL ASSETS
- AVIONIC TECHNOLOGIES
- SENSOR AND RADAR
- MOBILITY TECHNOLOGIES
- CRYPTOGRAPHIC SYSTEM SOLUTİONS
- NAVAL DEFENSE AND UNDERWATER ACOUSTICS
- ELECTRO-OPTICAL TECHNOLOGIES
- DIGITAL TRANSFORMATION AND SOFTWARE SOLUTIONS
- HIGH-POWER ENERGY LASER SYSTEMS

v240917

Business and **Collaboration Models**





BİLGEM Worldwide

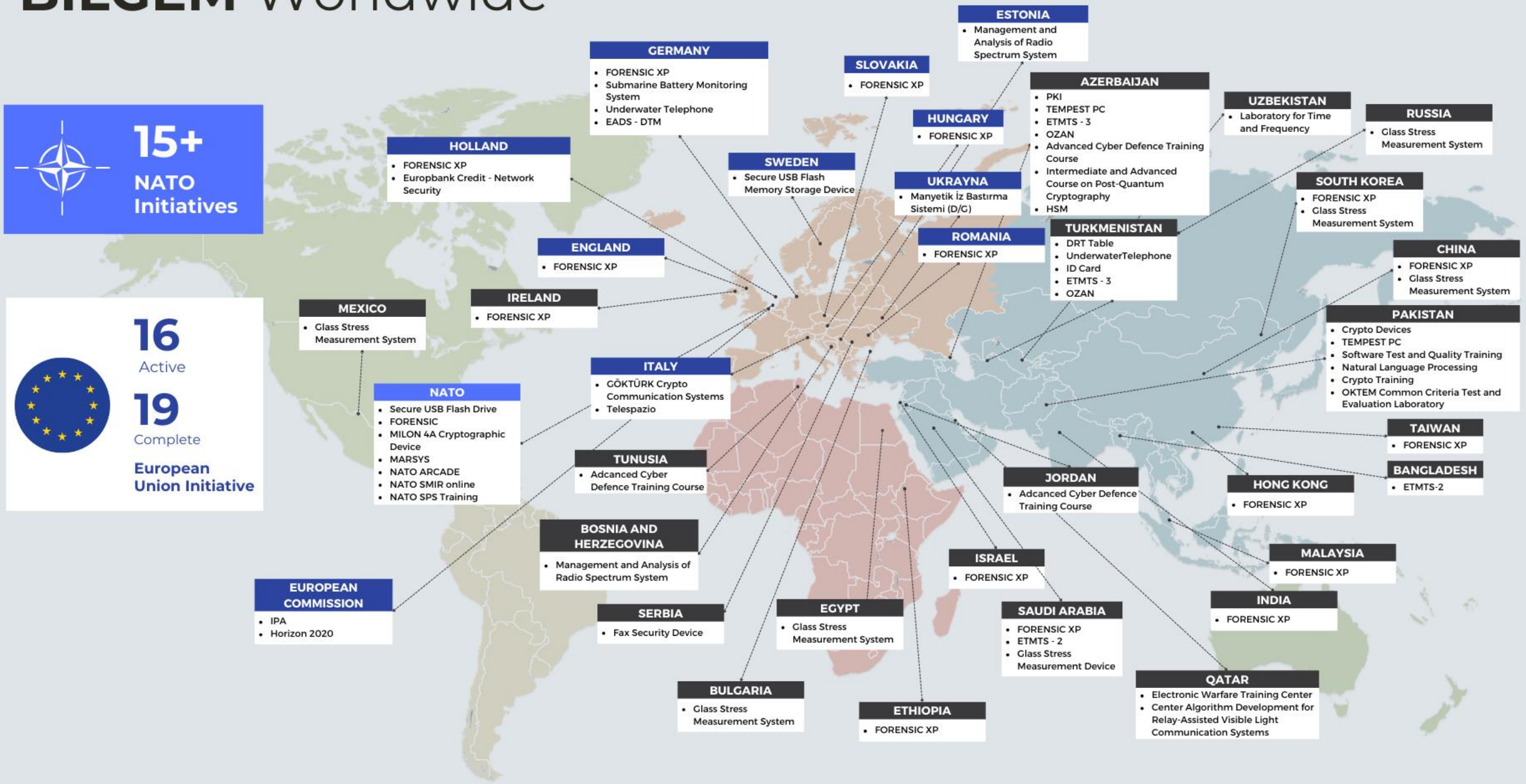
TÜBİTAK BİLGEM

15+
NATO Initiatives

16
Active

19
Complete

European Union Initiative



ESTONIA
- Management and Analysis of Radio Spectrum System

GERMANY
- FORENSIC XP
- Submarine Battery Monitoring System
- Underwater Telephone
- EADS - DTM

SLOVAKIA
- FORENSIC XP

AZERBAIJAN
- PKI
- TEMPEST PC
- ETMTS - 3
- OZAN
- Advanced Cyber Defence Training Course
- Intermediate and Advanced Course on Post-Quantum Cryptography
- HSM

UZBEKISTAN
- Laboratory for Time and Frequency

RUSSIA
- Glass Stress Measurement System

HUNGARY
- FORENSIC XP

HOLLAND
- FORENSIC XP
- Europbank Credit - Network Security

SWEDEN
- Secure USB Flash Memory Storage Device

UKRAYNA
- Manyetik İz Bastırma Sistemi (D/G)

SOUTH KOREA
- FORENSIC XP
- Glass Stress Measurement System

ROMANIA
- FORENSIC XP

TURKMENISTAN
- DRT Table
- UnderwaterTelephone
- ID Card
- ETMTS - 3
- OZAN

ENGLAND
- FORENSIC XP

CHINA
- FORENSIC XP
- Glass Stress Measurement System

MEXICO
- Glass Stress Measurement System

IRELAND
- FORENSIC XP

PAKISTAN
- Crypto Devices
- TEMPEST PC
- Software Test and Quality Training
- Natural Language Processing
- Crypto Training
- OKTEM Common Criteria Test and Evaluation Laboratory

ITALY
- GÖKTÜRK Crypto Communication Systems
- Telespazio

NATO
- Secure USB Flash Drive
- FORENSIC
- MILON 4A Cryptographic Device
- MARSYS
- NATO ARCADE
- NATO SMIR online
- NATO SPS Training

TAIWAN
- FORENSIC XP

BANGLADESH
- ETMTS-2

TUNUSIA
- Adcanced Cyber Defence Training Course

JORDAN
- Adcanced Cyber Defence Training Course

HONG KONG
- FORENSIC XP

BOSNIA AND HERZEGOVINA
- Management and Analysis of Radio Spectrum System

ISRAEL
- FORENSIC XP

MALAYSIA
- FORENSIC XP

INDIA
- FORENSIC XP

EUROPEAN COMMISSION
- IPA
- Horizon 2020

SERBIA
- Fax Security Device

EGYPT
- Glass Stress Measurement System

SAUDI ARABIA
- FORENSIC XP
- ETMTS - 2
- Glass Stress Measurement Device

BULGARIA
- Glass Stress Measurement System

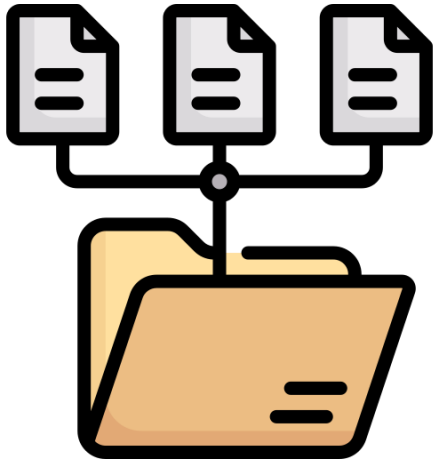
ETHIOPIA
- FORENSIC XP

QATAR
- Electronic Warfare Training Center
- Center Algorithm Development for Relay-Assisted Visible Light Communication Systems

INTRODUCTION

Paper Structure

1. Related Work
2. Methodology
3. Analysis
4. Conclusions and Future Work



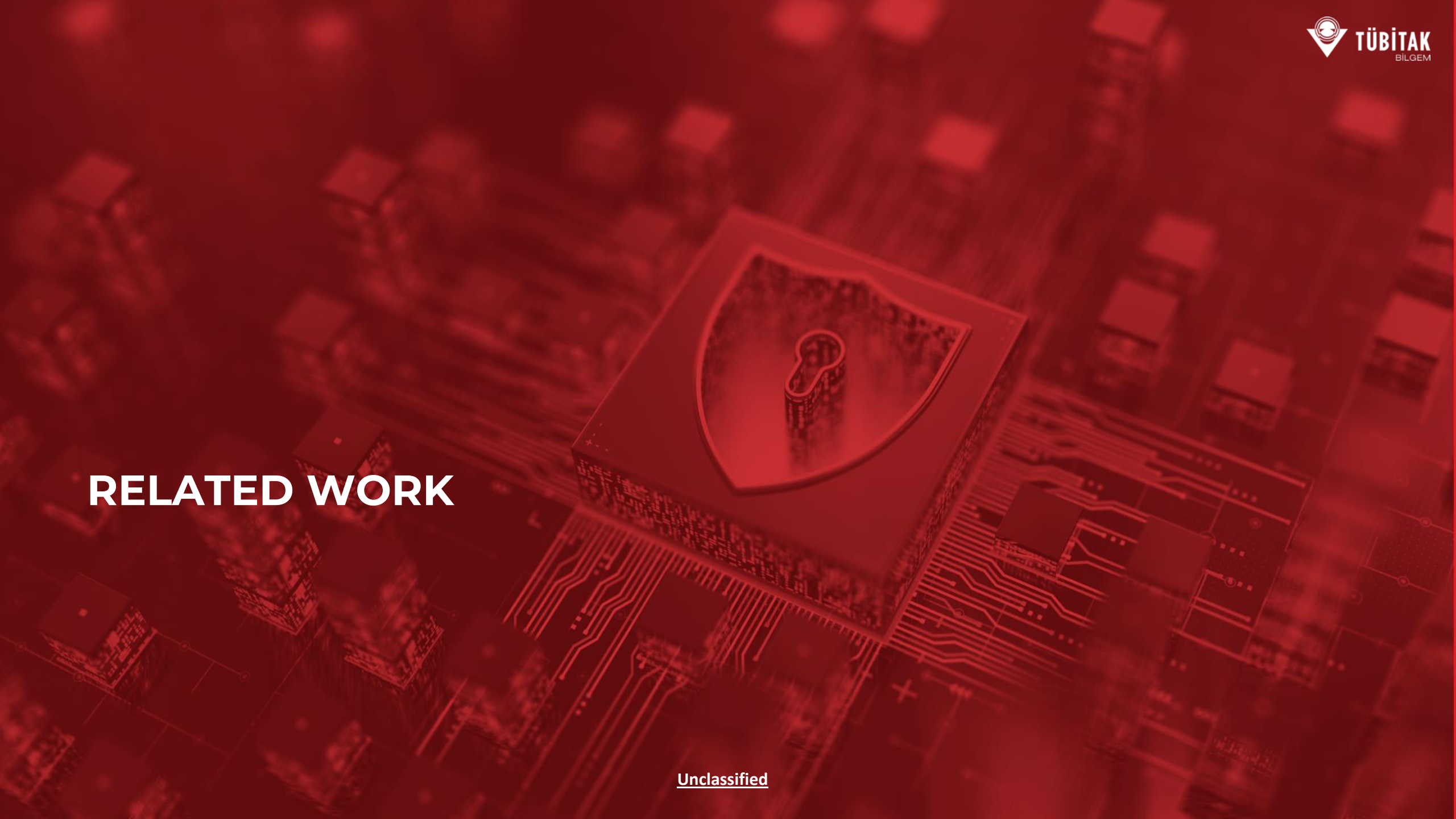
Why Software Defects Matter?

Why Are Software Defects Important?

- Defects (bugs/errors) directly impact reliability, maintainability, and user satisfaction.
- Early detection & resolution reduce rework, cost, and organizational risk.
- Crucial in public-sector projects where credibility and compliance are critical.
- Defect analysis enables prevention by tracing root causes.

What Makes the Public Sector Different?

- Greater **regulatory oversight** and **complex stakeholder** environments.
- **Longer** procurement and development **cycles**.
- Frequent **reuse** and **integration** across agencies → **higher defect** impact.
- Defects can **propagate** through **interconnected** systems.

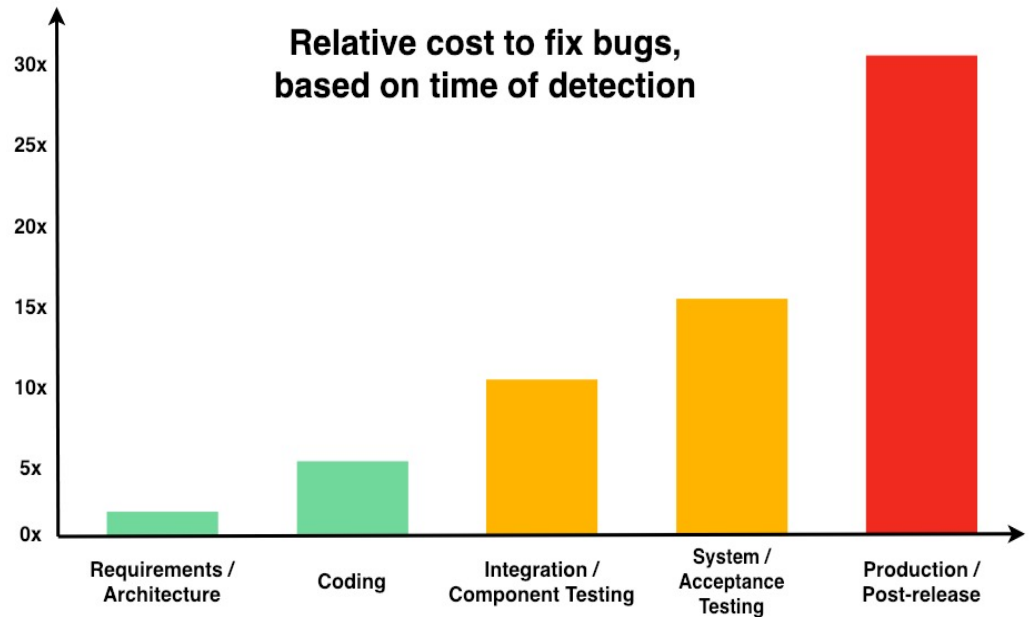


RELATED WORK

Importance of Defect Tracking in Software Quality

Why Defect Tracking Matters?

- Defect data is a critical management asset for process improvement (Grady, 1996).
- Defect frequency is a direct, quantifiable indicator of software quality.
- More defects = lower reliability, usability, efficiency.
- Early defect prevention is more cost-effective than late detection.

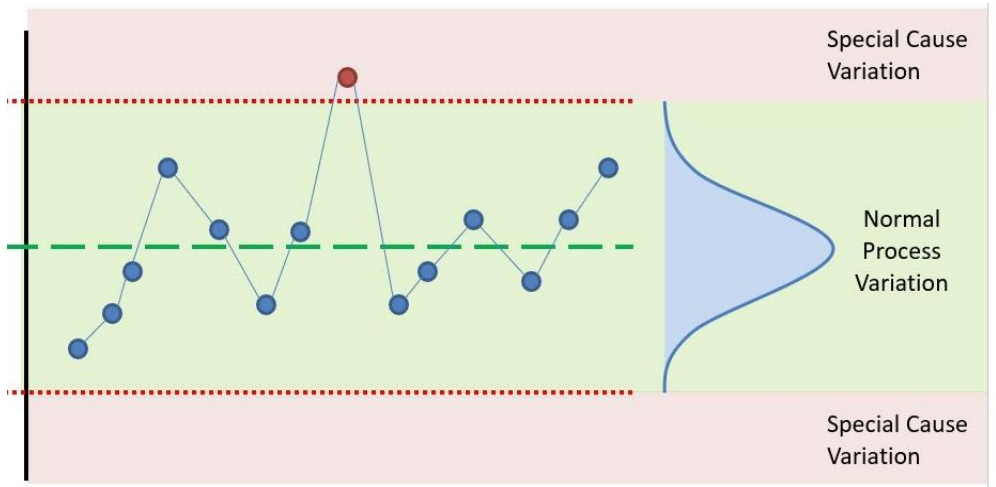


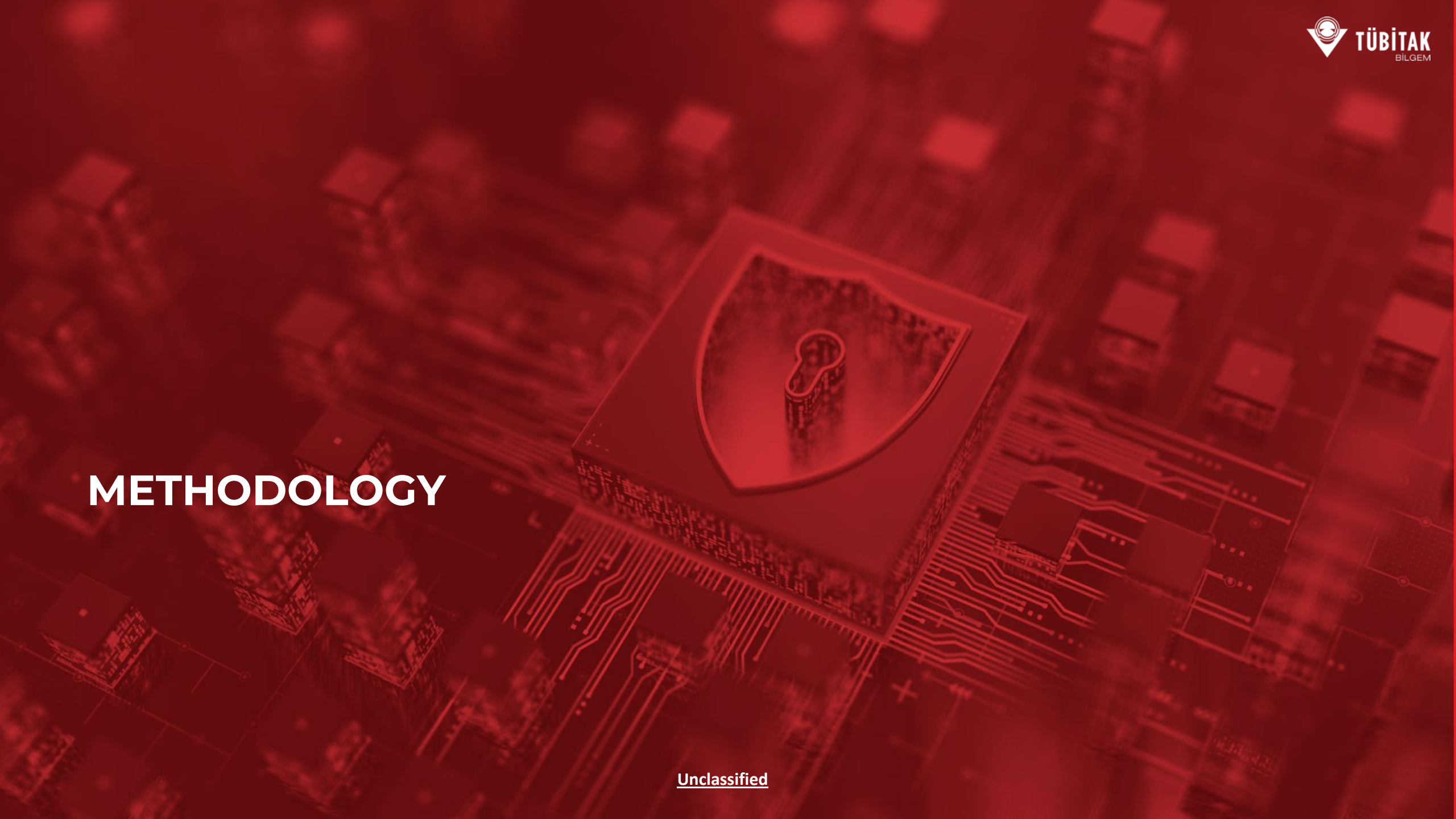
From Detection to Prevention: Defect Causal Analysis (DCA)

- DCA = structured method for **identifying root causes** of recurring defects.
- Uses **classification** tools like **Pareto** charts to **prioritize** frequent defect types.
- Enables targeted process improvements.
- Shifts focus from reactive fixing to proactive prevention.

Using Statistical Process Control (SPC)

- Software processes exhibit variation: natural (common) vs. assignable (special) causes.
- SPC helps distinguish between the two and identify instability in the process.
- Control charts (e.g., defect density over time) help monitor process health.
- Timely action on SPC data leads to sustained process improvement.





METHODOLOGY

Why This Study?

- Triggered by a noticeable increase in production defects in a public-sector software project.
- Aimed to **analyze patterns**, timing, and root causes of defects.
- R Studio dashboards signaled **unusual trends** → deeper investigation initiated.
- Scope refined with input from technical lead & project manager for relevance.
- Goal is to support shift from reactive defect handling to proactive quality assurance, aligned with TQM and CMMI.

Dataset Overview

Data Source and Structure

- Data pulled from a **task and issue management** platform used across the full SDLC.
- Platform captures **detailed metadata**: issue type, severity, component, status, etc.
- Dataset covers 147 resolved defects in production, from Jan 2024 to Apr 2025.
- Entries were validated & filtered in collaboration with project leadership.
- Focus: Only **confirmed production defects** (not test/staging environment issues).

Field	Value
Issue Key	143
Issue Type	Defect
Sprint Period	01.04.24
Severity	Medium
Defect Type	Coding
Component/s	A
Detected Activity	System Monitoring
Resolution	Done
Status	Closed

Key Variables Analyzed

Defect Variables Used in the Analysis

- **Severity**: Impact level of each defect (minor → major)
- **Detected Activity**: Phase where defect was found (e.g., Monitoring, Development, Test)
- **Component(s)**: Affected modules/subsystems
- **Detected Sprint**: When the issue was logged (enables time-based analysis)
- **Defect Type**: Technical nature (e.g., coding, data, UI)

Data Considerations and Limitations

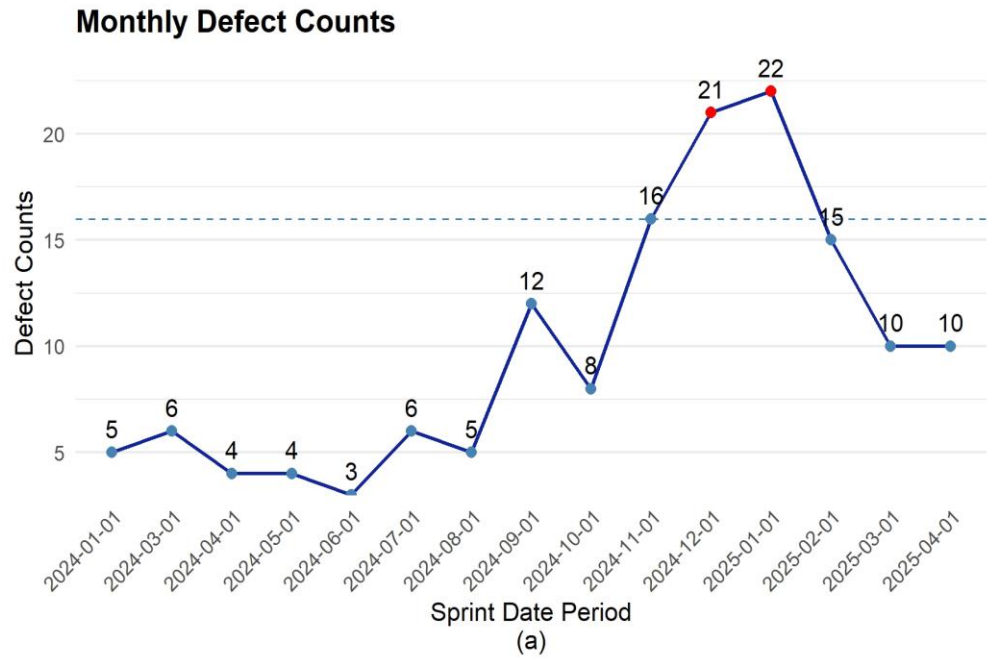
- **Manual data entry** may affect accuracy/timeliness.
- **Subjectivity** in issue classification (defect vs. other).Sample size = 147 defects, limits generalizability.
- Despite limitations, dataset provides **real-world traceability** and reflects **day-to-day dev operations**.
- Supported by literature: **Better data → Better defect prediction** models [1].

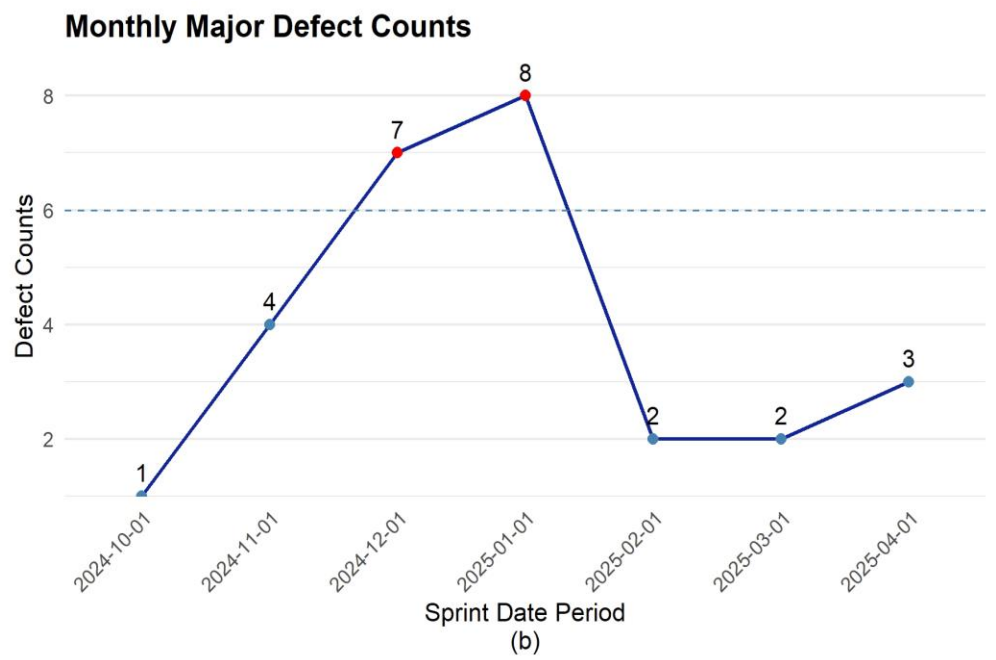
Expected Outcomes and Impact

- Identify conditions, components, and time periods with high defect incidence.
- Trace root causes of recurring issues using structured data.
- Reduce defect density → improves maintainability and user satisfaction.
- Support shorter release cycles, lower maintenance costs, and greater trust in public services.

ANALYSIS

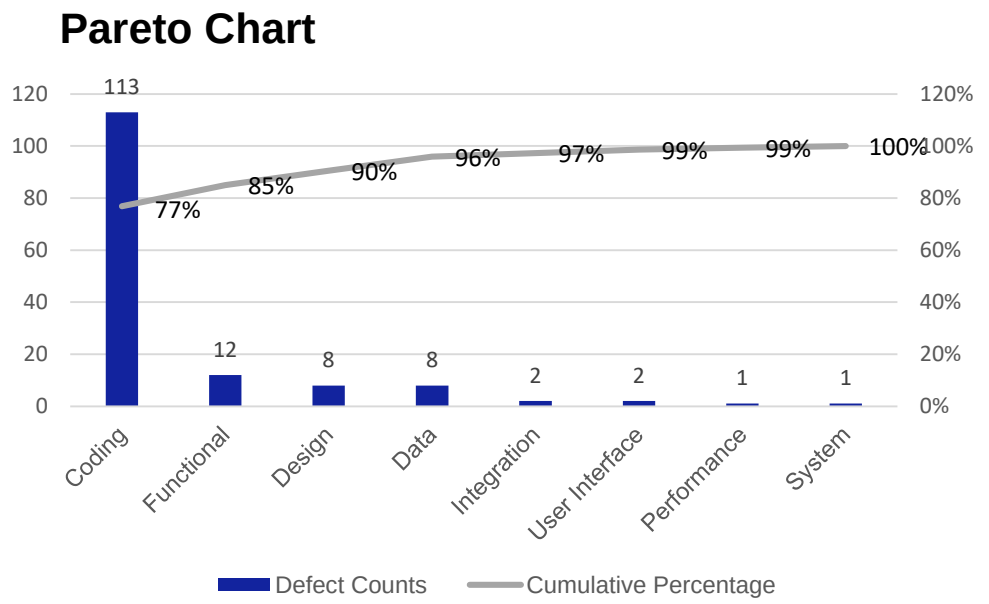
Monthly Defect Counts

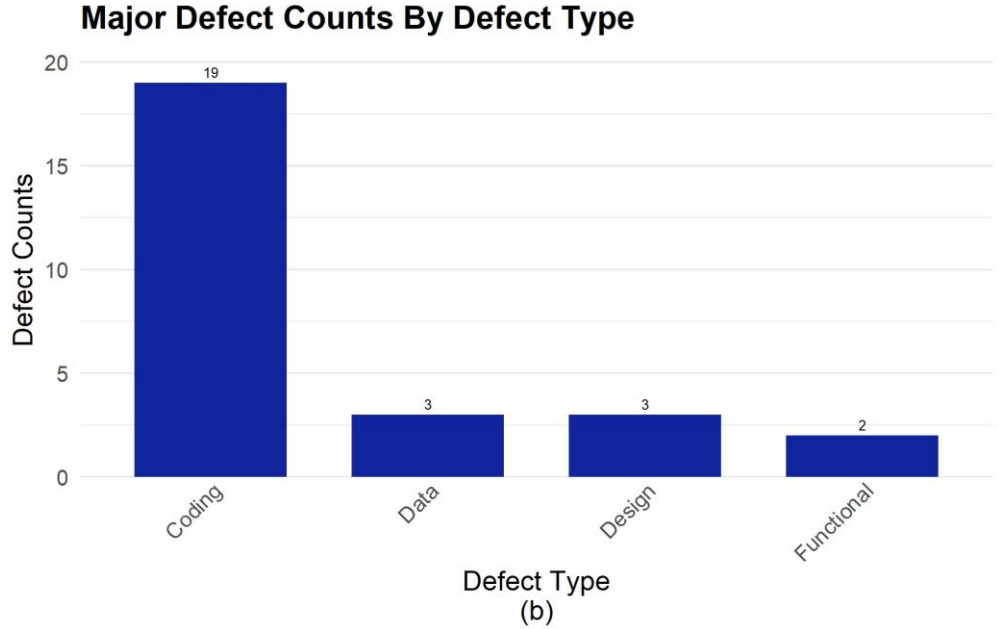




- Total and major defect counts increased significantly in some months.
- UCL (mean + 1σ) used to flag statistically significant anomalies.
- December 2024 and January 2025 exceeded control limits.
- Root cause: likely due to changes in reporting behavior, not true defect spikes

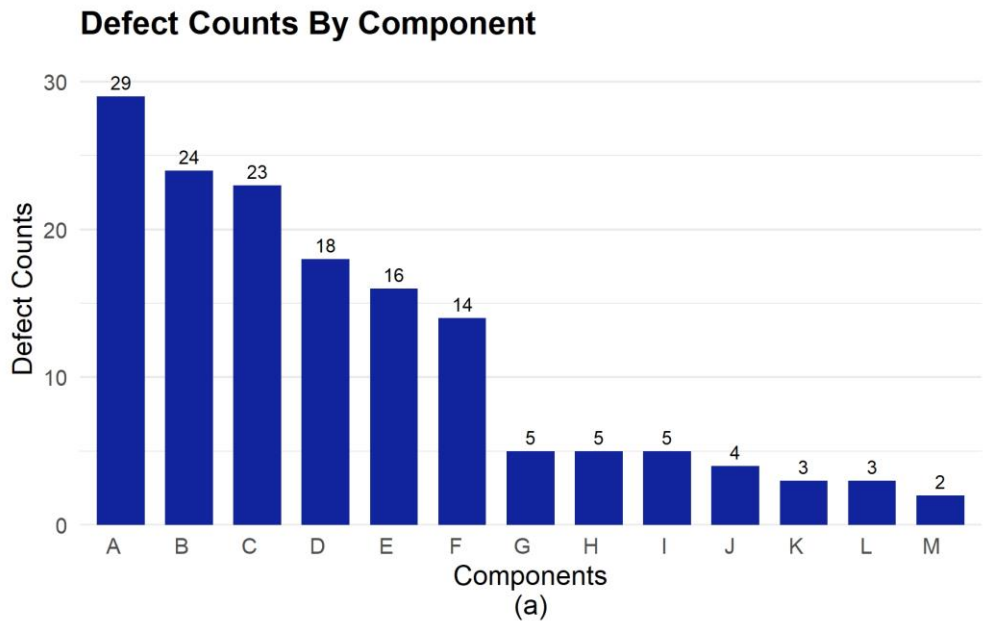
Defect Counts by Defect Type

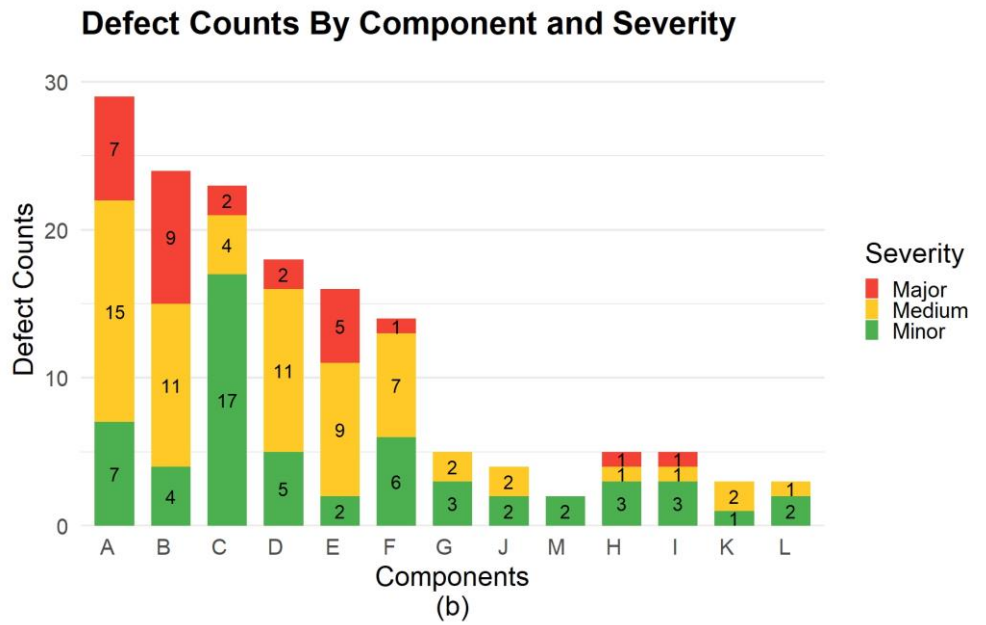




- Coding defects = 77% → largest opportunity for quality improvements.
- Other frequent types: functionality, architecture, data.
- Most major defects also stem from coding issues.
- Findings support focus on developer training and code reviews

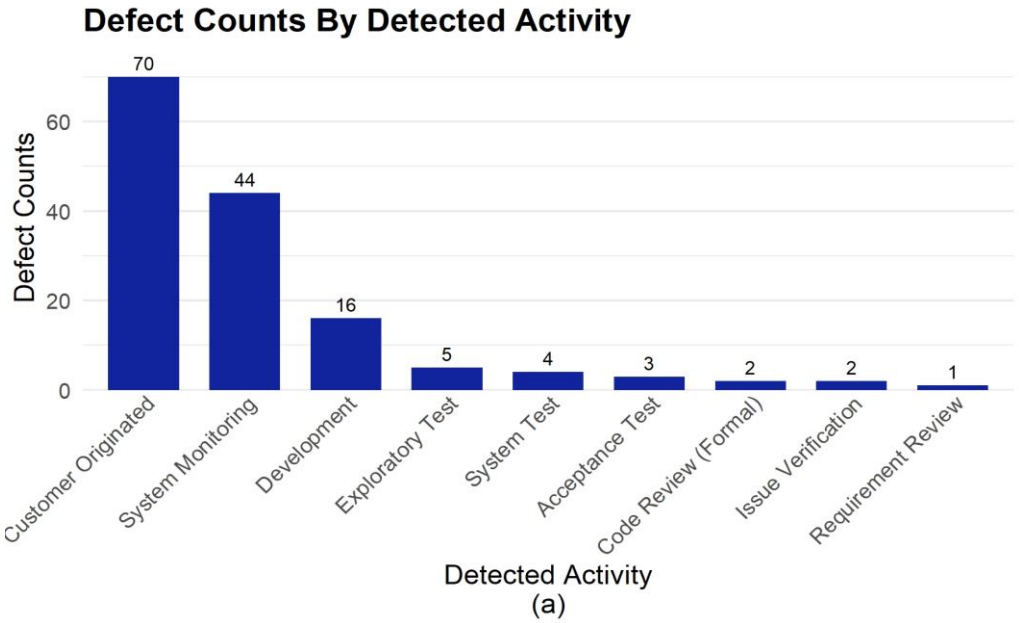
Defect Counts by Component

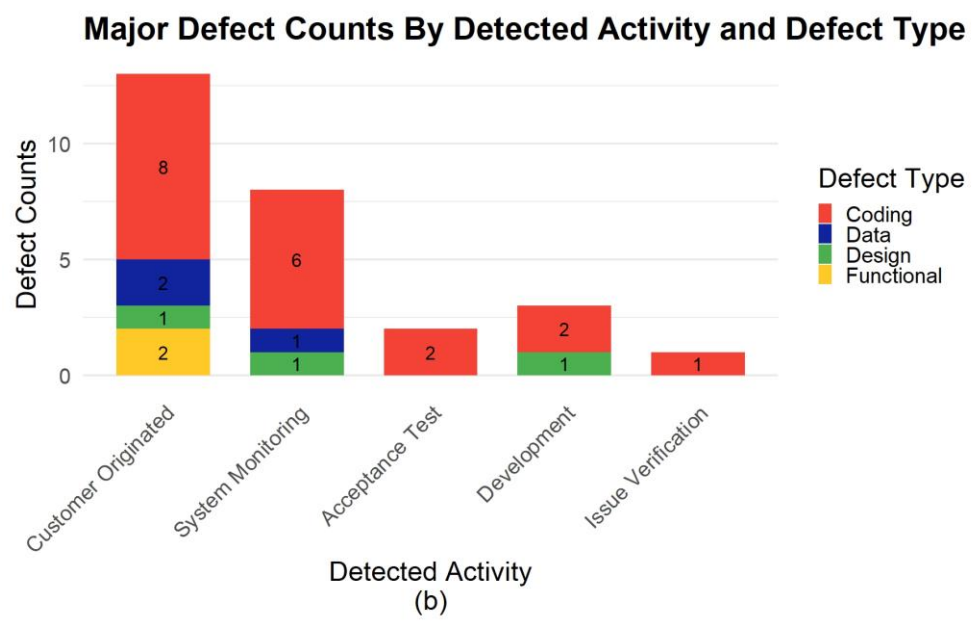




- Three components show notably high defect concentration.
- Component E: lower total defects but high % of major issues.
- Component C: most defects, but majority are minör.
- Insight helps prioritize quality actions per component

Defect Counts by Detected Activity

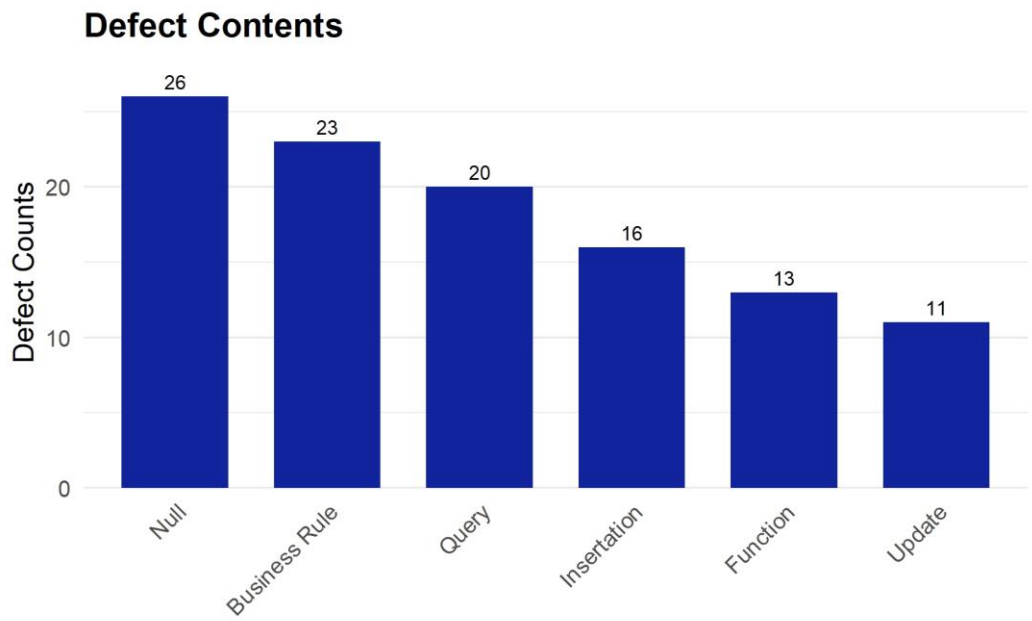




- Most defects detected post-deployment, especially by customers (70 cases).
- System monitoring tools caught 44 issues.
- Indicates room for improvement in early testing phases.
- Early detection reduces customer impact and operational risk

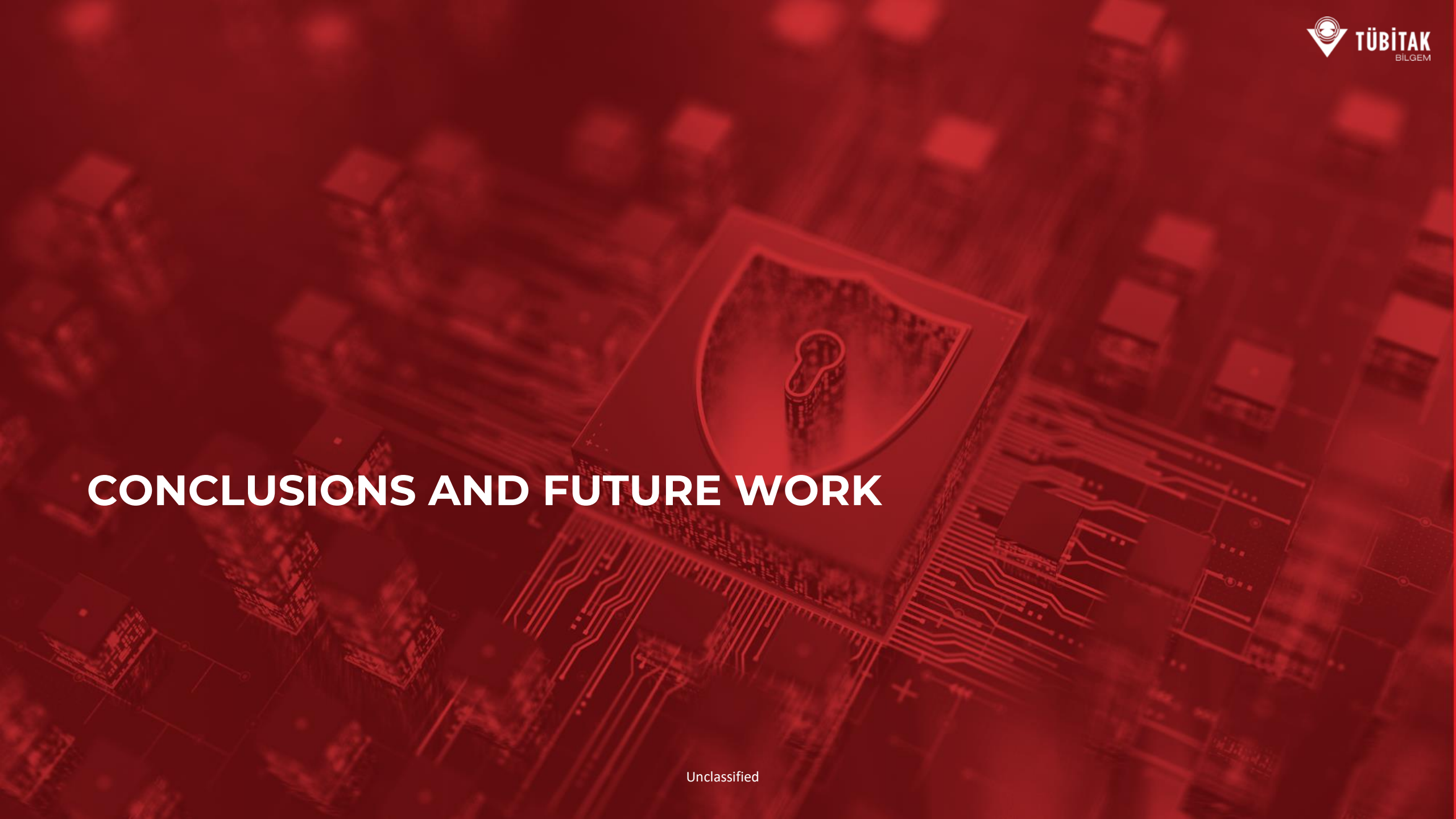
Analysis of Defect Issue Summarries

- Top defect types:
    - Null Pointer Exceptions (NPEs) – 26
    - Business rule violations – 23
    - Query issues – 20
- NPEs often caused by dynamic data sources (beyond static analysis).
- Suggest training, improved input validation, and business rule alignment.



Key Findings

- Coding defects dominate → prioritize code quality practices.
- High post-release detection → improve test coverage & early QA.
- Address reporting consistency through training or input validation.
- Analyze business rule violations for process misalignments



CONCLUSIONS AND FUTURE WORK

Conclusions

Key Conclusions

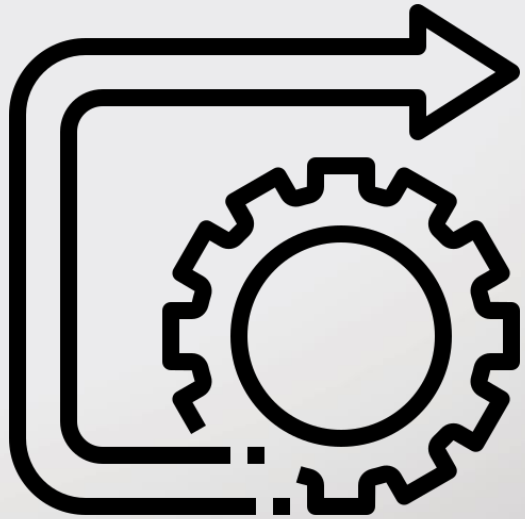
- Coding defects are the dominant issue, especially related to Null Pointer Exceptions (NPEs).
- Most major defects were Customer Originated, highlighting late detection.
- Components A and B are the most defect-prone in both total and major categories.
- Frequent business rule violations suggest weaknesses in requirement analysis.

Recommendations

- Improve test coverage beyond happy paths, including edge cases (e.g. null handling).
- Conduct targeted developer training on code robustness and NPE prevention.
- Investigate root causes of business rule violations (customer vs. analyst errors).
- Increase analyst involvement in early project stages.

Future Work

- Shift from reactive to proactive defect management.
- Integrate additional quality metrics for continuous monitoring.
- Develop a predictive model to anticipate high-risk defects pre-deployment.



REFERENCES

References

[1] Bosu, M.F., & MacDonell, S.G. (2013). Data quality in empirical software engineering: a targeted review. In: Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering (EASE 2013), pp. 171–176.

REPUBLIC OF TÜRKİYE
MINISTRY OF INDUSTRY
AND TECHNOLOGY

#NATIONAL TECHNOLOGY INITIATIVE

TÜBİTAK
BİLGEM

15 YEARS
AT THE HEART OF
TECHNOLOGY

Thank you