An NSF-awarded Digital Engineering Tool for Integrated System Reliability Assessment

Kazuhira Okumoto, Ph. D., CEO
Sakura Software Solutions, LLC

RAMS 2026

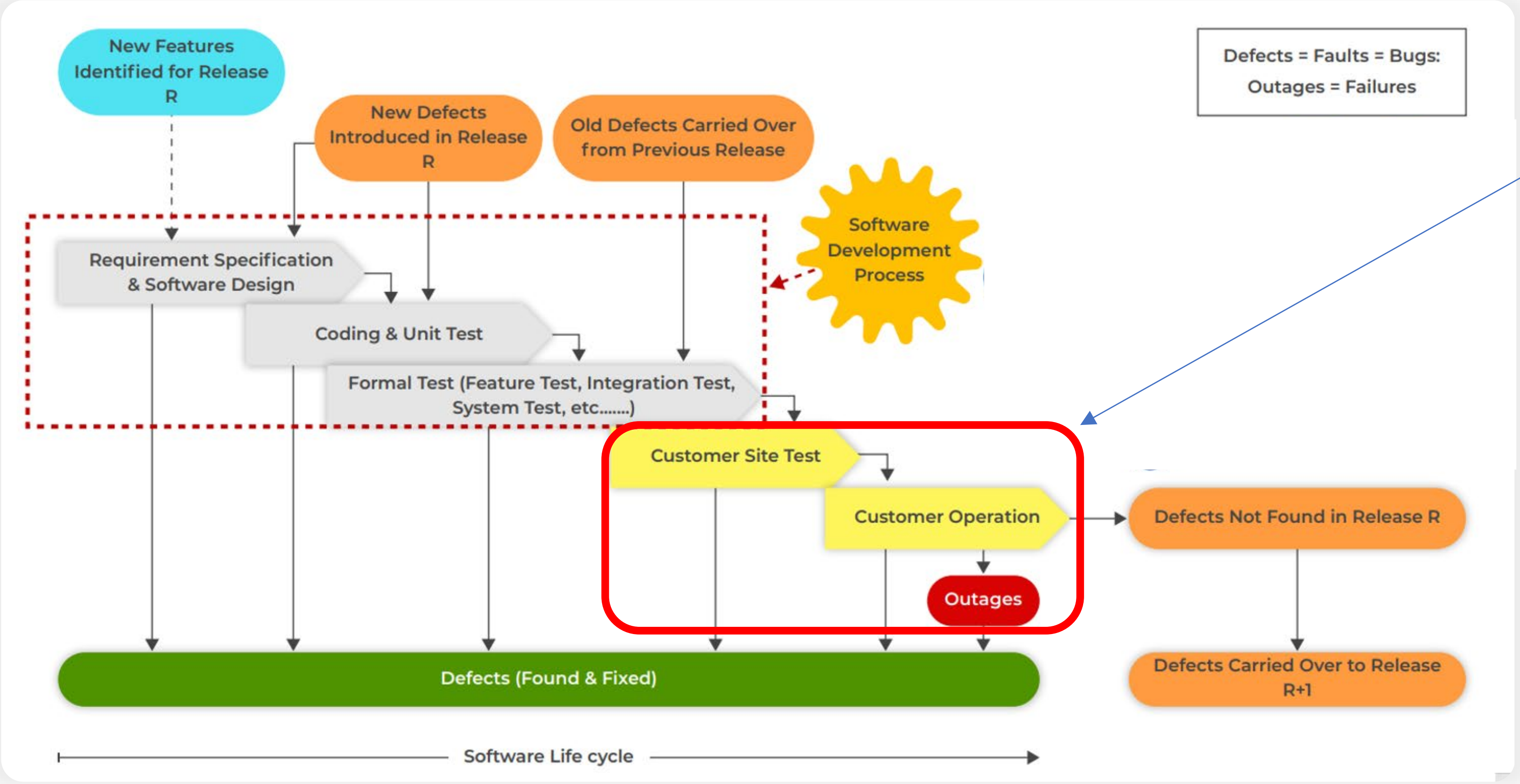
Overview and Outline

- Background and Introduction
- FUSION Overview
- Constant Software Failure Rate
- Intelligent Graphical Editor (iGRED) for Reliability Block Diagrams (RBD)
- Summary & Conclusions
- Next Steps and Future Work

Background and Introduction

- Software Development Process vs. Defect Injection & Removal -

Software operational failures account for 60–80% of system issues.



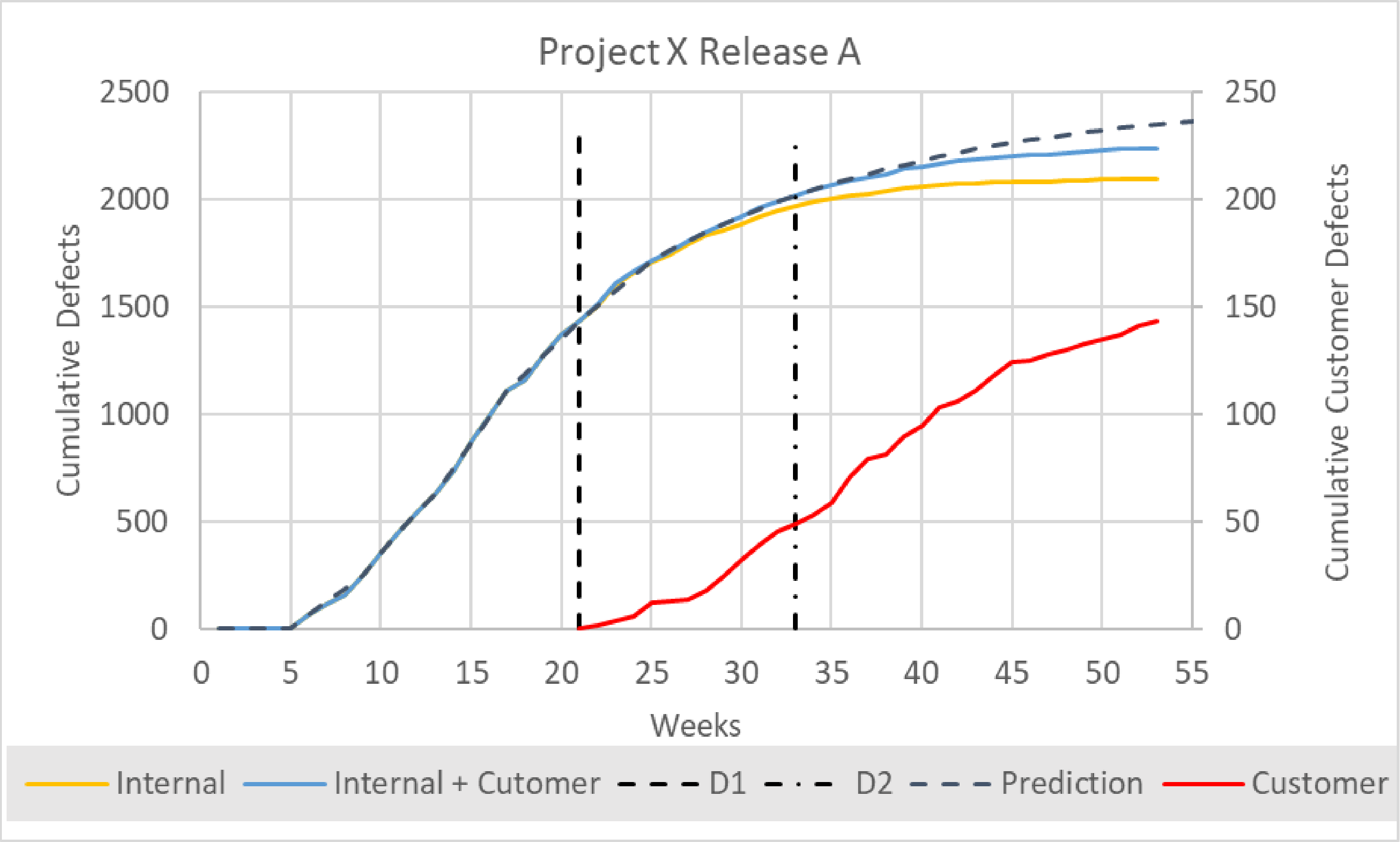
Gap in current approaches

Background and Introduction

- Software defect curves: Internal vs. Customer-found defects -

Can we use customer-found defects to demonstrate a constant software failure rate?

Answering this question has direct implications for **how we model, predict, and improve software quality**



Overview and Outline

- Background and Introduction
- FUSION Overview
- Constant Software Failure Rate
- Intelligent Graphical Editor (iGRED) for Reliability Block Diagrams (RBD)
- Summary & Conclusions
- Next Steps and Future Work

FUSION Overview

- A Digital Engineering Tool for Integrated Software and Hardware Reliability -

The Challenge

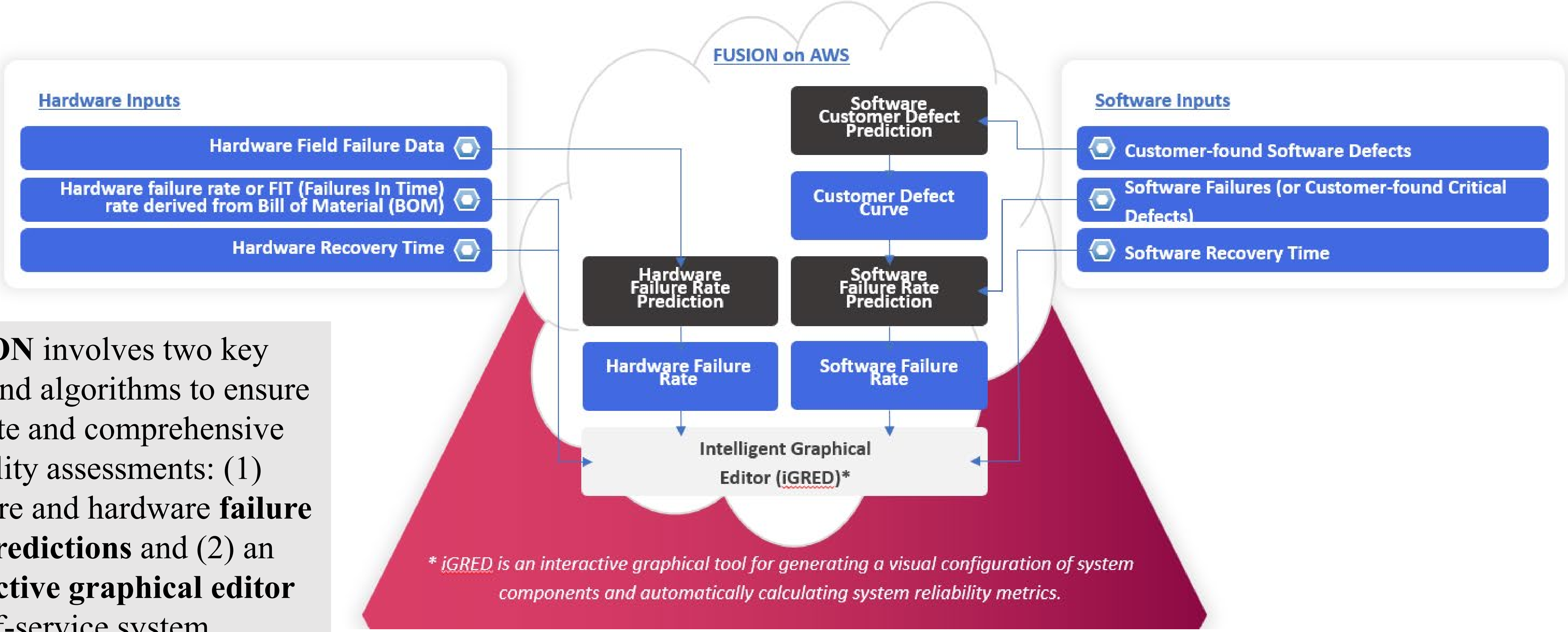
- Customer-found defects do not follow constant failure rates
- Traditional hardware models fail to capture software failure dynamics
- Misaligned software-hardware reliability leads to inaccurate system evaluations
- Existing tools lack integrated analysis across software and hardware components

The Solution: FUSION

- Integrates software and hardware reliability in a unified cloud-based platform
- Models non-constant software failure rates using advanced analytics
- Features intelligent, interactive Reliability Block Diagrams (RBDs)
- Designed for system integrators and service providers managing complex systems

FUSION Overview

- A Digital Engineering Tool for Integrated Software and Hardware Reliability -

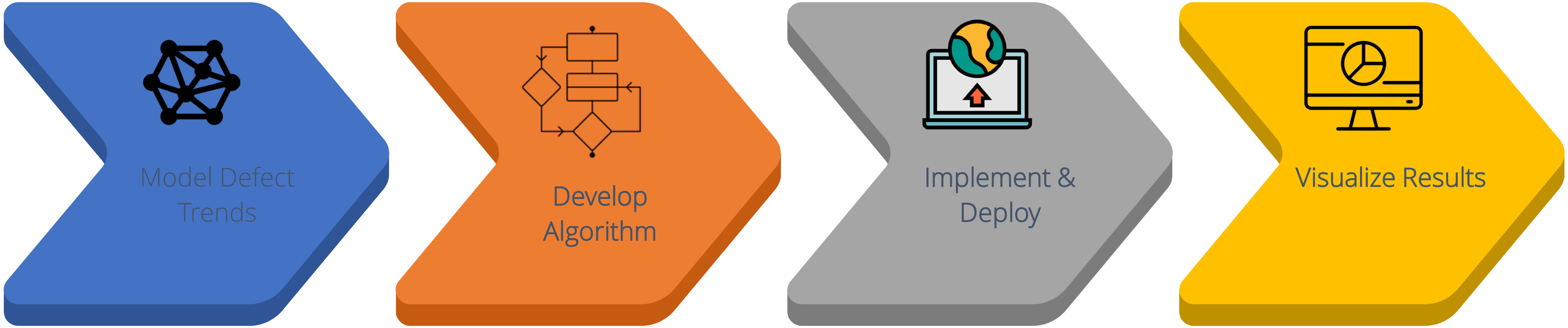


FUSION involves two key steps and algorithms to ensure accurate and comprehensive reliability assessments: (1) software and hardware **failure rate predictions** and (2) an **interactive graphical editor** for self-service system reliability.

Overview and Outline

- Background and Introduction
- FUSION Overview
- Constant Software Failure Rate
- Intelligent Graphical Editor (iGRED) for Reliability Block Diagrams (RBD)
- Summary & Conclusions
- Next Steps and Future Work

Customer Defect Prediction



- We assume **multiple straight lines** can describe defect trends. See the chart.

- Algorithm to:
  - Automatically identify key turning points (**inflection points**)
  - Generate a series of straight lines from these points

- Implement the algorithm in **Python** and port it to the cloud.

- Create clear, visualized output of the predictions.

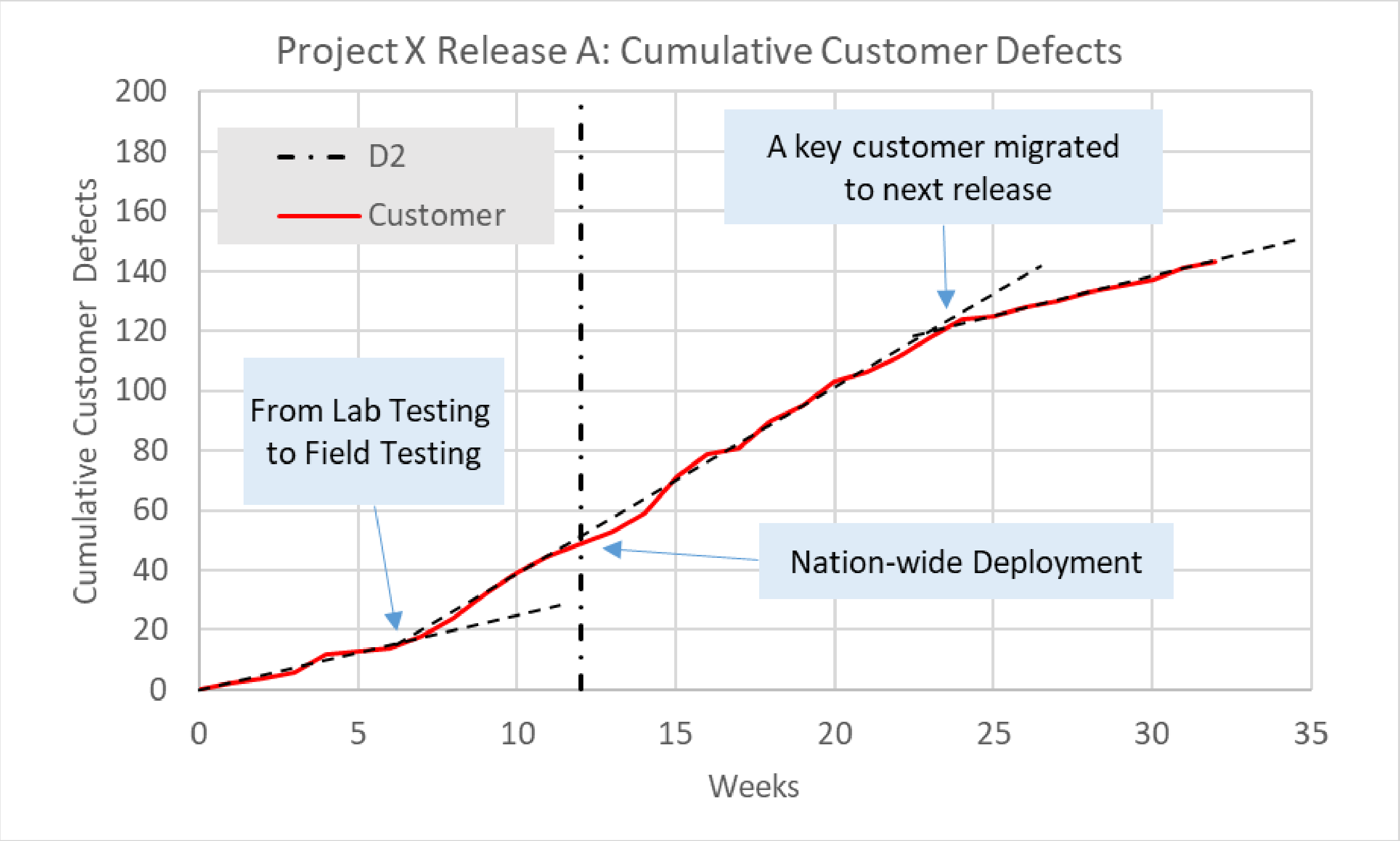
Validate Effectiveness: Verify the algorithm's effectiveness using **multiple datasets**.

Customer Defect Prediction

Piece-wise linear modeling reveals inflection points tied to **deployment events**, enhancing defect trend predictions.

$$m(x) = m(x_{j-1}) + \theta_j \, (x - x_{j-1})$$

for $x_{j-1} \leq x \leq x_j$

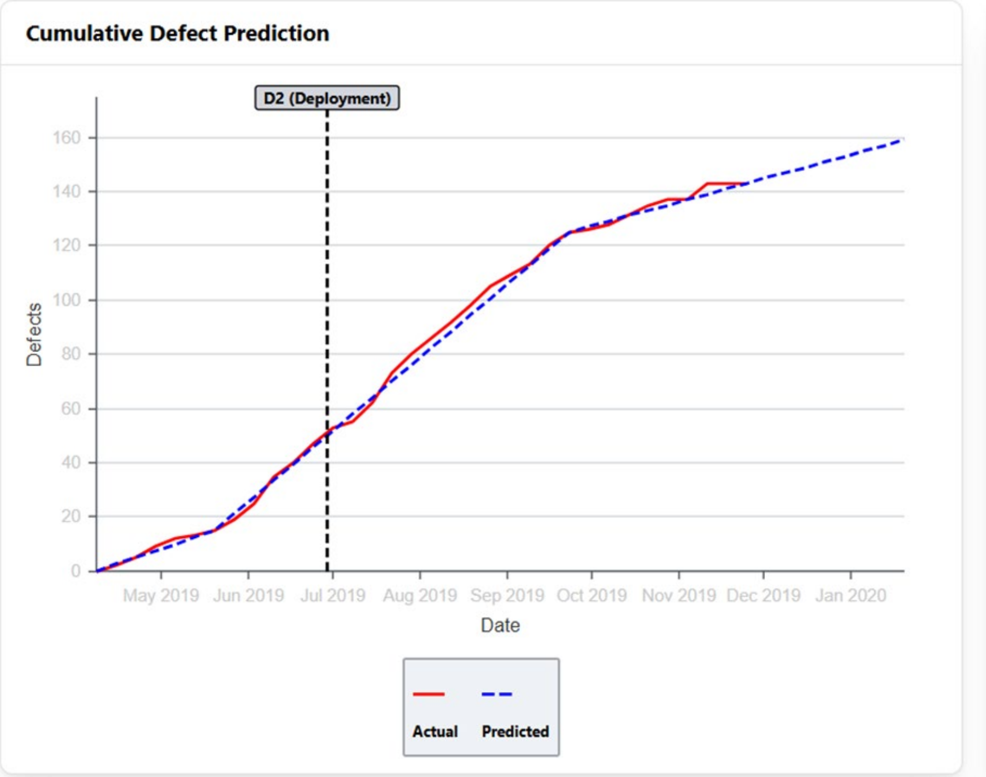


Customer Defect Prediction – Sample Output

- Fusion automatically identifies inflection points to generate multiple straight lines to describe the entire defect trend

Cumulative Defect Prediction

- **Fusion's model closely tracks real defect accumulation over time**, validating prediction accuracy across deployment phases.
- **Piece-wise linear segments reveal trend shifts**, offering insights into reliability changes post-deployment.

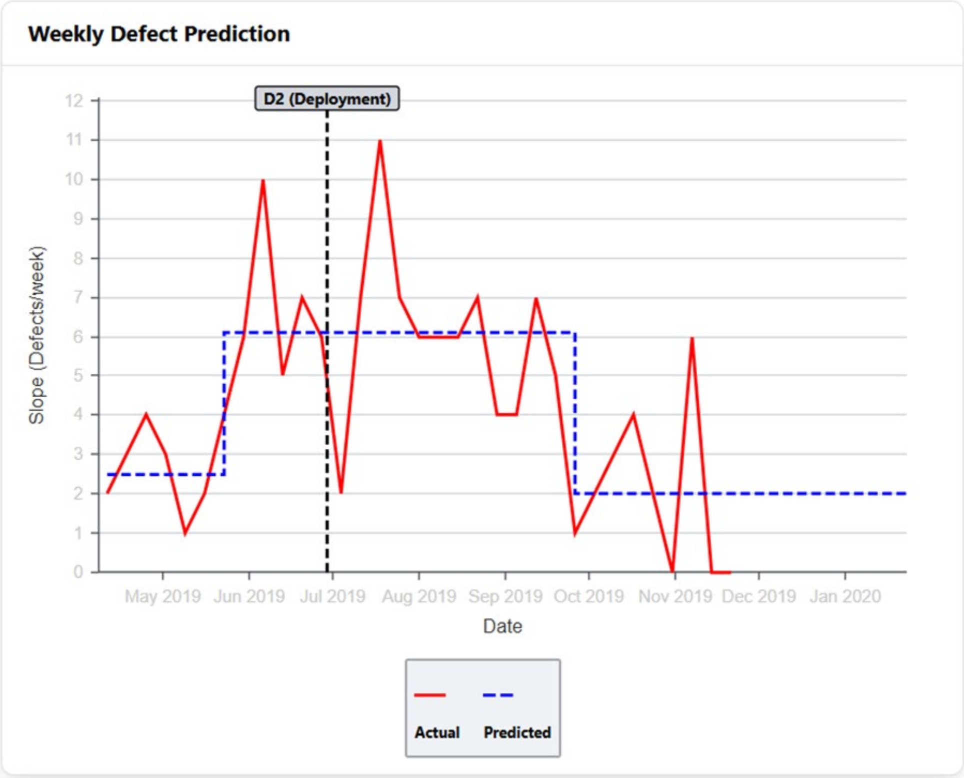


Customer Defect Prediction – Sample Output

- Fusion automatically identifies inflection points to generate multiple straight lines to describe the entire defect trend

Weekly Defect Prediction

- **Fusion identifies stable periods of defect generation**, smoothing out noisy weekly variations in actual data.
- **Inflection-based prediction captures defect rate shifts**, enabling proactive quality and maintenance decisions.



Software Failure Rate Prediction

1 Model

We assume the **defect curve** can describe the **failure curve** since failures are a subset of defects.

2 Algorithm

Automatically find the failure curve to **minimize the difference** between actual and transformed curves.

3 Implement

Implement the algorithm in **Python** and port it to the cloud.

4 Visualize

Create clear, visualized output of the predictions.

5 Validate

Verify the algorithm's effectiveness using multiple datasets.

The **transformation function** will effectively map the defect curve into the failure curve. See the chart.

$$n(u) = f(m(x))$$

$$n(u) = n(u_{j-1}) + \lambda_j (u - u_{j-1}) \quad \text{for } u_{j-1} \le u \le u_j$$

Transformation function:

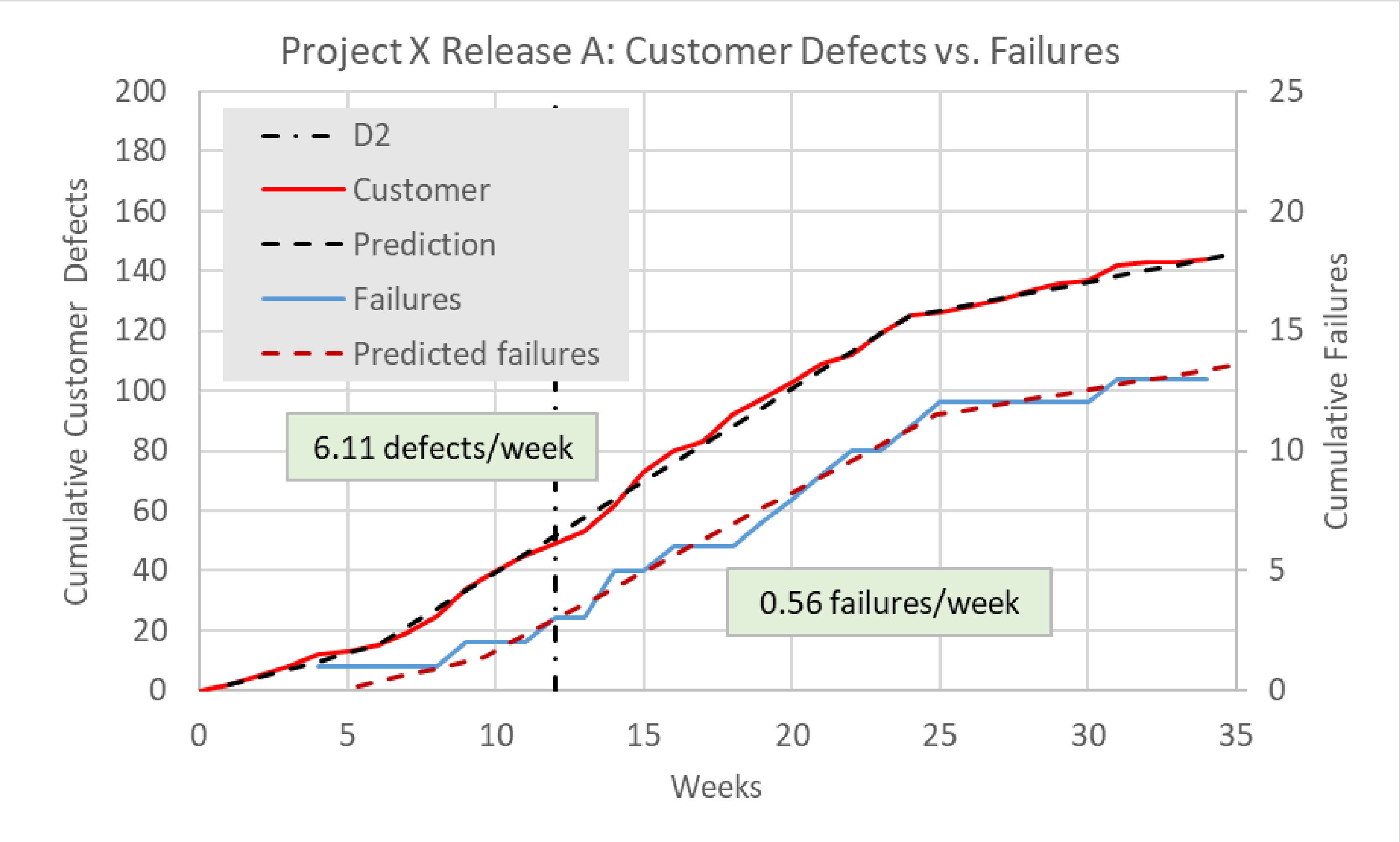
$$u = \alpha + \beta x$$

$$n(u) = \gamma\, m(x)$$

Software Failure Rate Prediction

Customer Defects vs. Failures

- **Fusion accurately models both defect and failure accumulation,** capturing real-world divergence between software issues and system-level failures.
- **Predictive insights into failure rates (0.56/week) vs. defect rates (6.11/week)** enable better prioritization and reliability planning



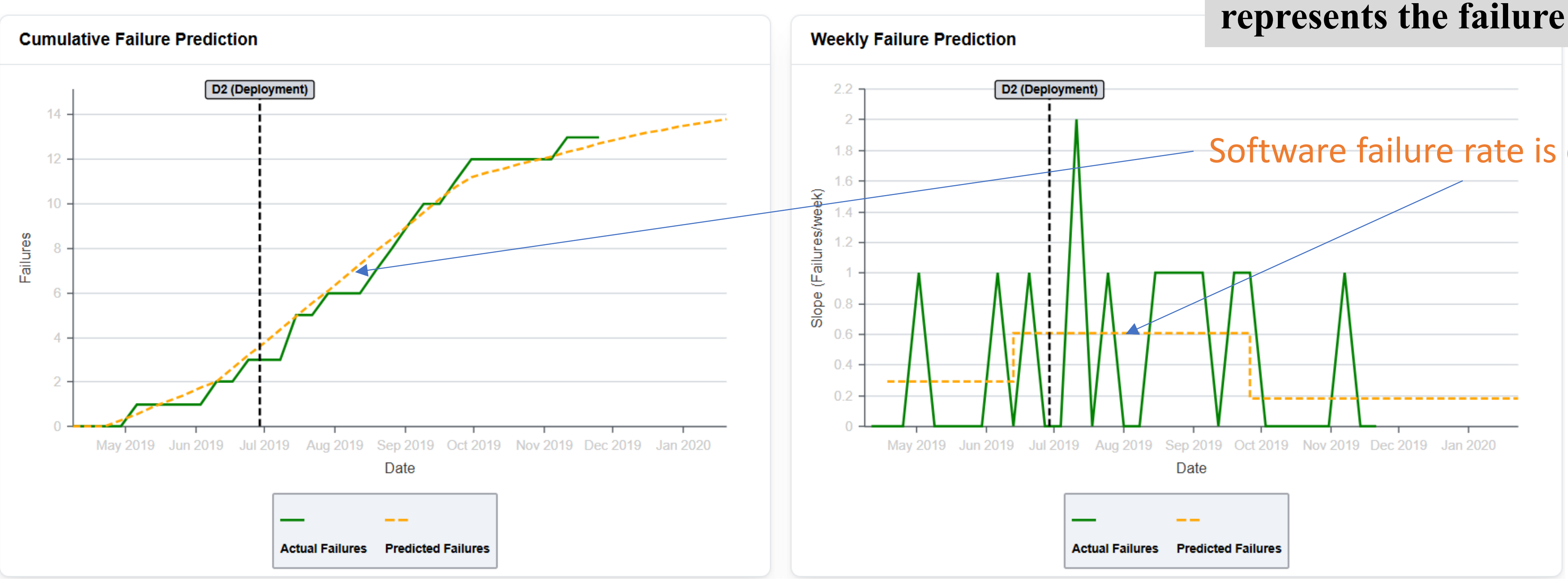
Note: Failures = "Critical" software defects

Software Failure Rate Prediction – Sample Output

- FUSION automatically converts the predicted customer defect curve into a software failure curve using a transformation function

The slope right after the deployment represents the failure rate

The software failure rate at deployment is 0.58 failures/week or 30.2 failures/year. ?

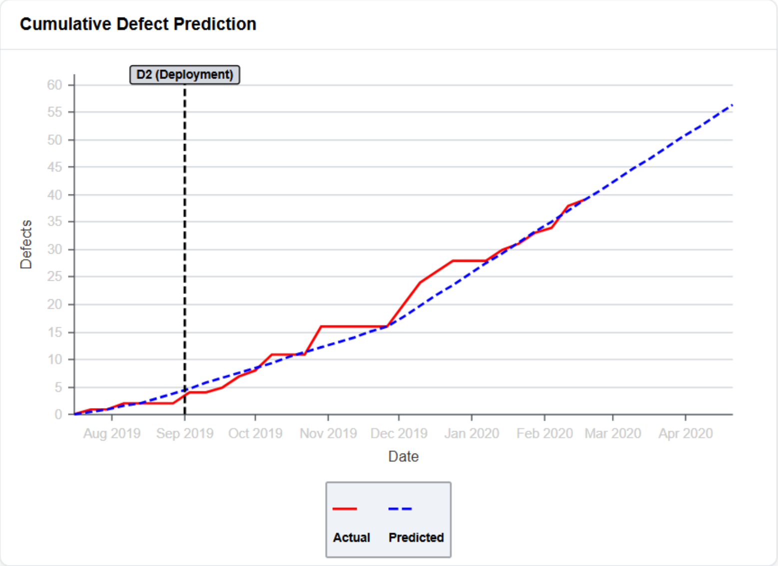


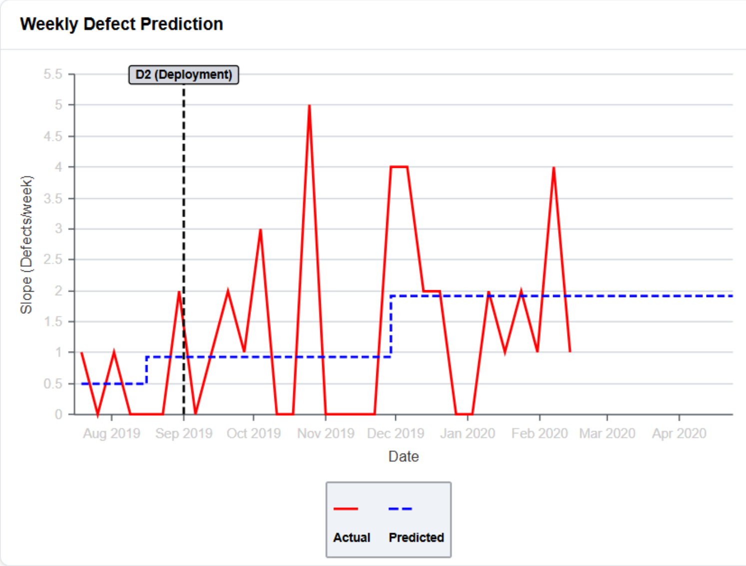
Software failure rate is constant!

Software Failure Rate Prediction – A Smaller Project Example

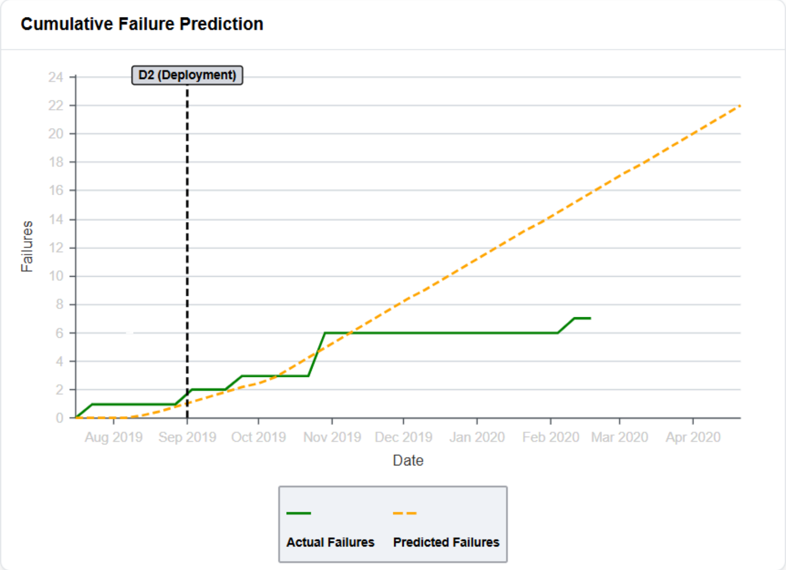
The software defect rate at deployment is 0.93 defects/week or 48.6 defects/year.

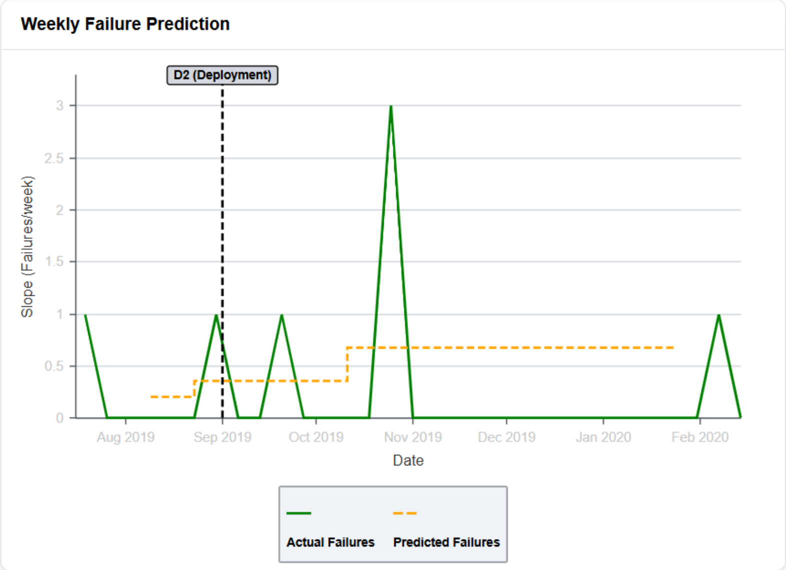
- **Fusion accurately predicts both defect and failure rates**, even in projects with limited data and low volume.
- **Inflection-based modeling remains effective**, capturing subtle trend shifts across deployment events.





ment is 0.34 failures/week or 17.7 failures/year.





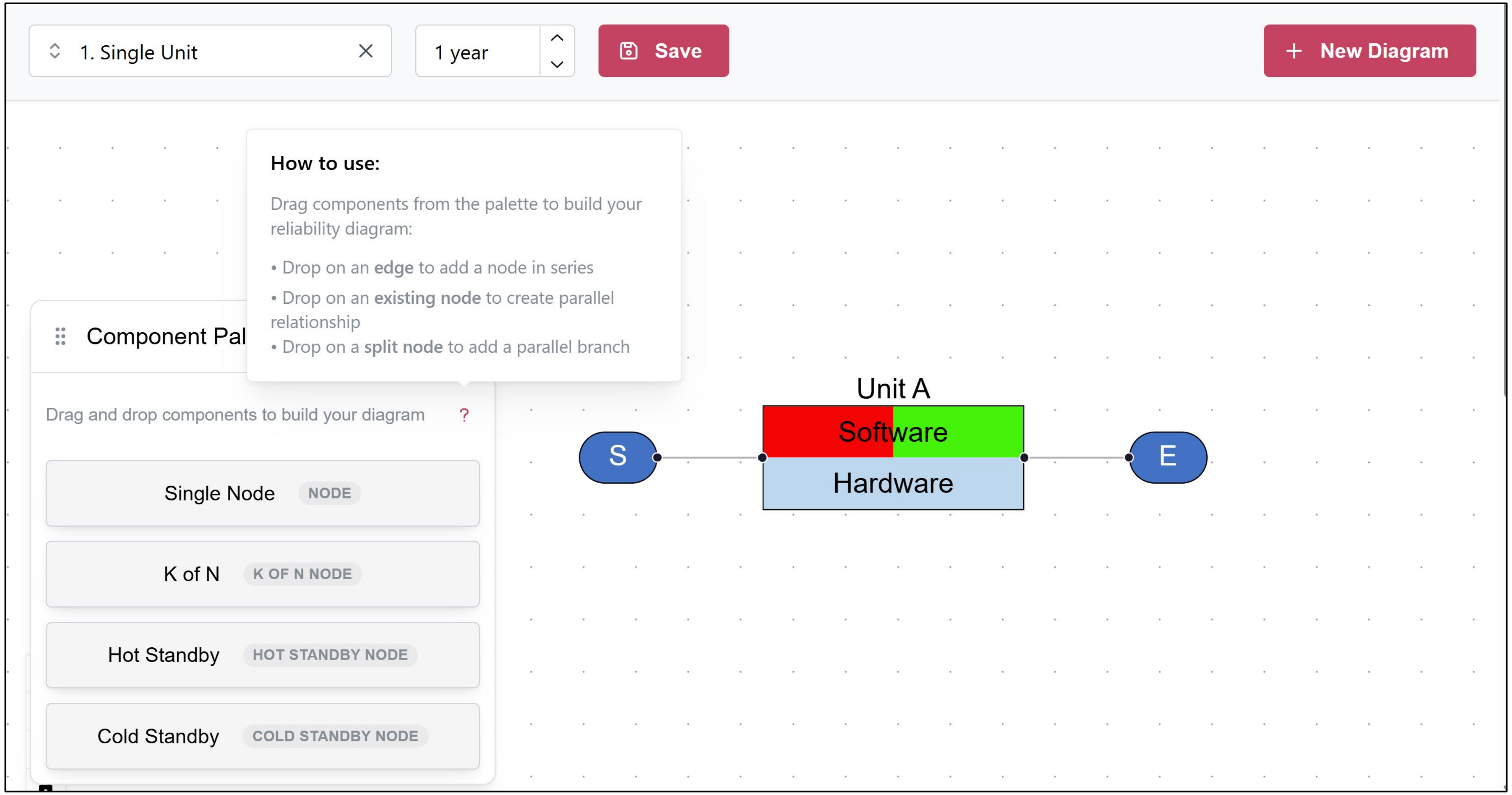
Overview and Outline

- Background and Introduction
- FUSION Overview
- Constant Software Failure Rate
- Intelligent Graphical Editor (iGRED) for Reliability Block Diagrams (RBD)
- Summary & Conclusions
- Next Steps and Future Work

Intelligent Graphical Editor (iGRED) for Reliability Block Diagrams (RBDs)

1. Interactive creation of RBDs for software and hardware systems with pre-configured templates (single, parallel, series, k-out-of-n)

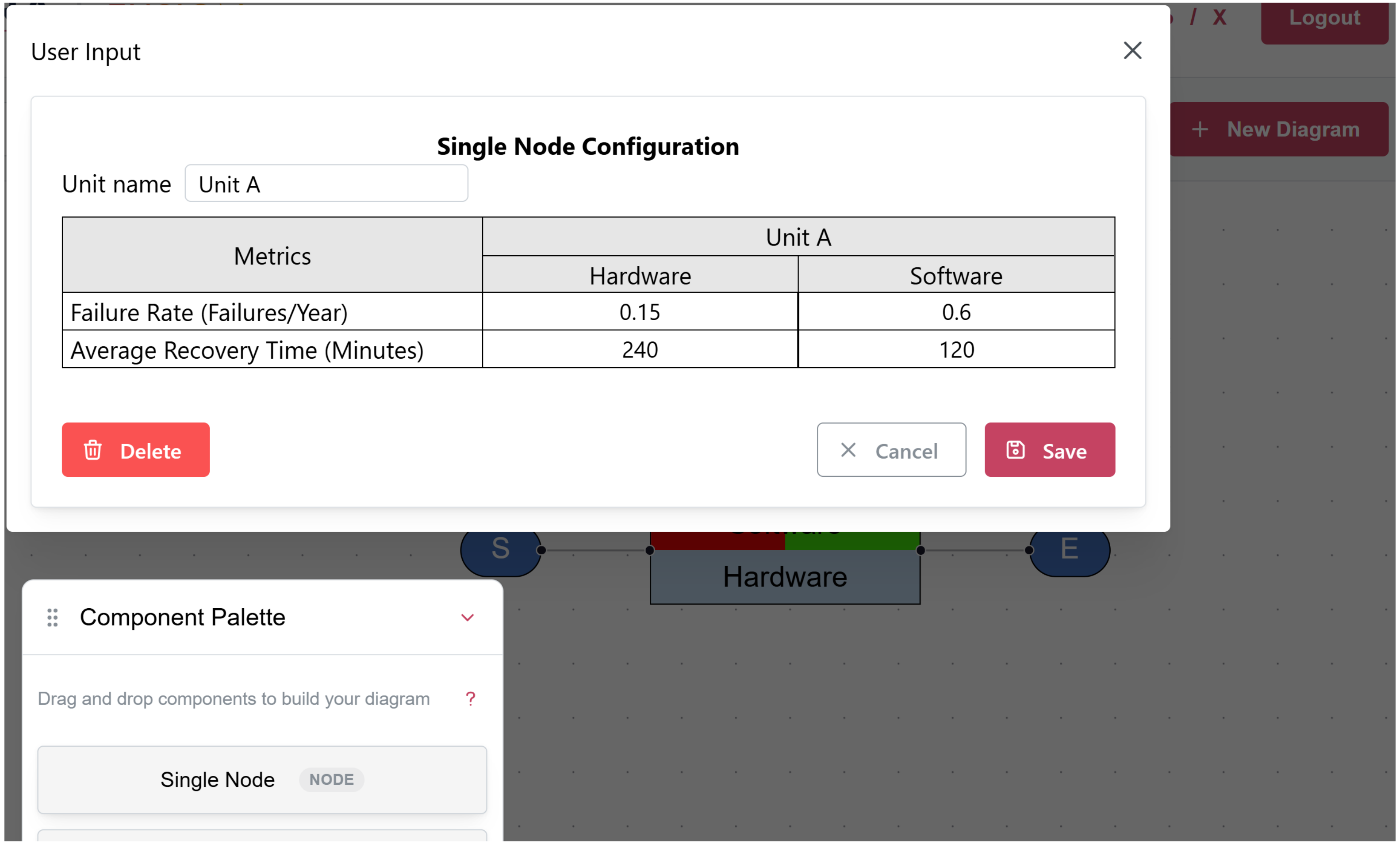
After creating a new diagram name, drag and drop the diagram template into the space below by following the "How to use" instructions.



Intelligent Graphical Editor (iGRED) for Reliability Block Diagrams (RBDs)

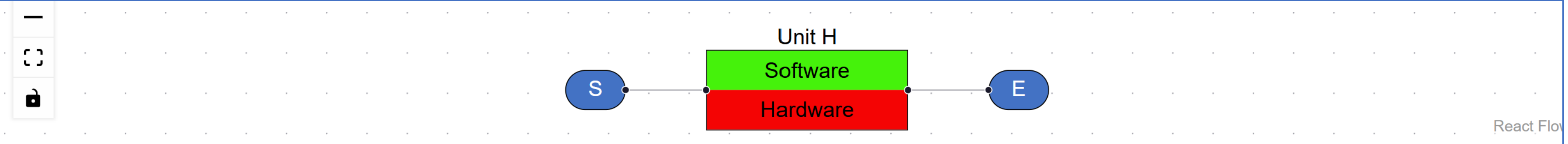
2. User inputs are: Failure rate and mean repair time for software and hardware

Clicking a node will open a pop-up window for user input.



Intelligent Graphical Editor (iGRED) for Reliability Block Diagrams (RBDs)
- Built-in real-time reliability metric calculations: Failure rate, MTTF, Reliability, MTTR, MTBF, Availability)

Save the diagram and click the Reliability Metrics Summary menu. The system will display the reliability metrics table and the bottleneck nodes.



Case H

ADVANCED NODE DETAILS FORMULAS

Bottleneck Nodes:

Availability: Unit H - hardware (Availability: 99.772%)

MTTF: Unit H - software (MTTF: 0.3333 Years/Failures)

Reliability Metrics	Unit	Hardware	Software	System (SW+HW)
Failure Rate	*Failures/Year*	2.000	3.000	5.000
MTTF (Mean Time To Failure)	*Years/Failure*	0.5000	0.3333	0.2000
Reliability at one year	*Percentage*	13.5	5.0	0.7
MTTR (Mean Time To Repair)	*Minutes/Repair*	600.000	200.000	360.000
MTBF (Mean Time Between Failures)	*Years/Failure*	0.5011	0.3337	0.2007
Availability	*Percentage*	99.772	99.886	99.659

Intelligent Graphical Editor (iGRED) for Reliability Block Diagrams (RBDs)
- Sample Metrics Formulas -

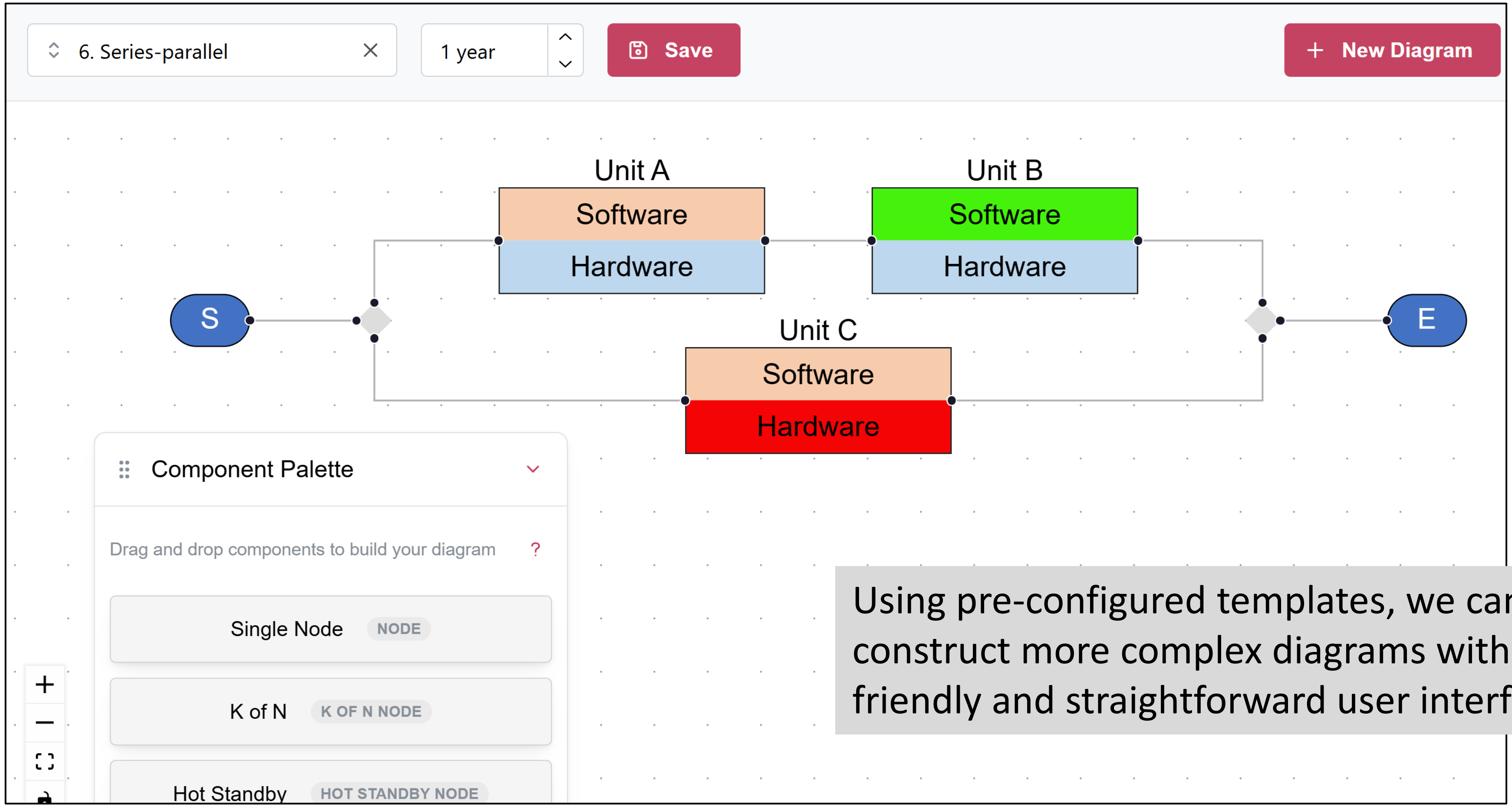
Clicking "Show Metrics Formulas" will open a new window for formulas.

A Single-Unit Configuration

Metric	Formulas		
	Hardware	Software	System
Failure Rate	λ_{hw}	λ_{sw}	$\lambda_{hw} + \lambda_{sw}$
MTTF	$\frac{1}{\lambda_{hw}}$	$\frac{1}{\lambda_{sw}}$	$\frac{1}{\lambda_{sys}}$
Reliability	$e^{-\lambda_{hw}t}$	$e^{-\lambda_{sw}t}$	$e^{-\lambda_{sys}t}$
MTBF	$MTTF_{hw} + MTTR_{hw}$	$MTTF_{sw} + MTTR_{sw}$	$MTTF_{sys} + MTTR_{sys}$
MTTR	$\frac{1}{\mu_{hw}}$	$\frac{1}{\mu_{sw}}$	$\frac{1}{\mu_{sys}} = (\frac{\lambda_{hw}}{\mu_{hw}} + \frac{\lambda_{sw}}{\mu_{sw}})/\lambda_{sys}$
Availability	$\frac{MTTF_{hw}}{MTBF_{hw}}$	$\frac{MTTF_{sw}}{MTBF_{sw}}$	$\frac{MTTF_{sys}}{MTBF_{sys}}$

Intelligent Graphical Editor (iGRED) for Reliability Block Diagrams (RBDs)

- Interactive creation of RBDs for software and hardware systems with pre-configured templates (single, parallel, series, k-out-of-n) – More complex example



Using pre-configured templates, we can construct more complex diagrams with a friendly and straightforward user interface.

Overview and Outline

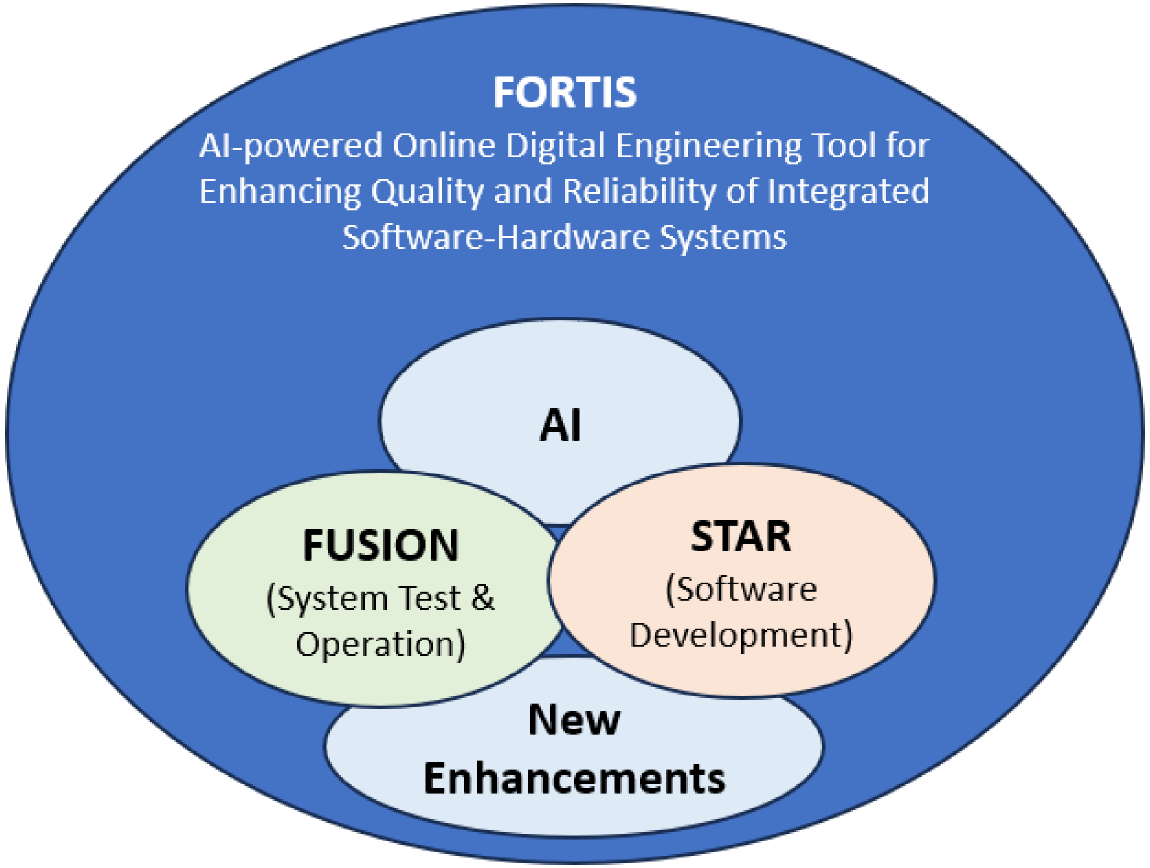
- Background and Introduction
- FUSION Overview
- Constant Software Failure Rate
- Intelligent Graphical Editor (iGRED) for Reliability Block Diagrams (RBD)
- Summary & Conclusions
- Next Steps and Future Work

Summary & Conclusions

- **Introduced FUSION — a groundbreaking, cloud-native digital engineering platform developed with support from the National Science Foundation.**
- FUSION seamlessly integrates **real-world software failure behavior** with **hardware reliability models**, moving beyond static lookup tables to data-driven accuracy.
- Through **interactive reliability block diagrams (RBDs)** and **automated predictive analytics**, FUSION delivers **actionable, system-level insights** on reliability, availability, and performance — empowering engineers to make informed decisions across complex software–hardware ecosystems.

Next Steps and Future Work
-AI-powered Online Digital Engineering Tool

An AI-powered digital engineering solution, **FORTIS**, will unify the capabilities of STAR (software quality assurance) and FUSION (system reliability analysis) into a single, end-to-end platform.



FORTIS Target Customers
- FUSION: System integrators and service providers
- STAR: Software development organizations

References

- K. Okumoto, "Digital Engineering-driven Software Quality Assurance-as-a-Service," in Proc. 28th ISSAT Int. Conf. Reliability and Quality in Design (RQD), 2023.
- H. Pham, Software Reliability, Springer, Berlin, Heidelberg, 1999.
- K. Trivedi, Reliability and Availability of Hardware-Software Systems: Stochastic Reliability Models of Real Systems, 2021, 10.13140/RG.2.2.30286.48960.
- Relyence, "A Deep Dive into System Modeling Using Reliability Block Diagram (RBD) Analysis," Technical report, 2021.