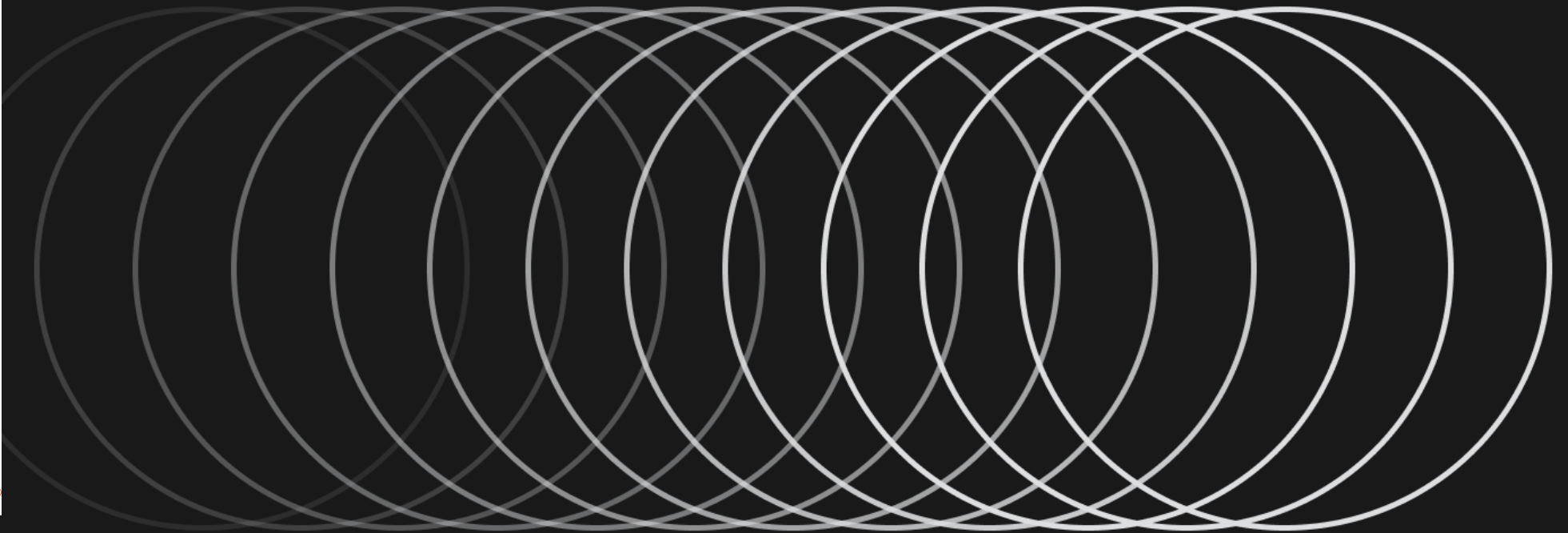


Predict software defects and technical debt training

Learn how to predict defects early and shift-left for optimal quality assurance in agile development environments.



Agenda

01. Shift-Left Advantage

Understanding the new era of quality assurance and proactive defect management methodologies.

02. Core Methodology

Learning the step-by-step process from risk assessment to defect arrival time forecasting.

03. Quantitative Metrics

Mastering predictive analytics for testing defects, technical debt, and resource allocation planning.

04. Target Audience

Identifying who benefits most from defect prediction skills and implementation strategies.



The Shift-Left Quality Revolution

Proactive Quality Management

Transform your approach to software quality by predicting defects before code is written. This course empowers teams to manage release stability, performance, and safety through advanced prediction methodologies designed specifically for agile development environments.



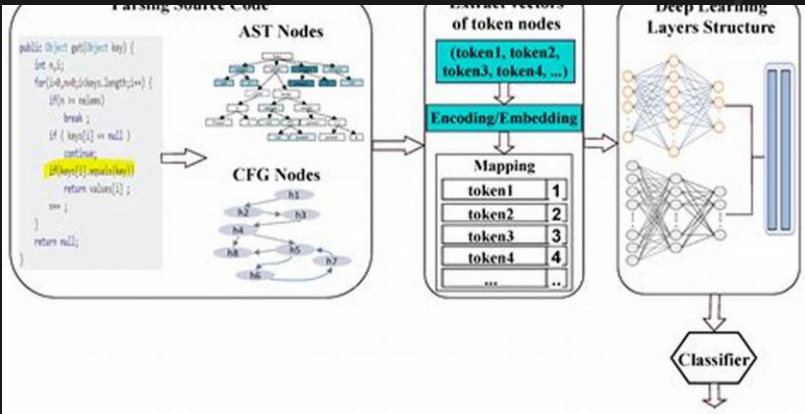
Risk Prevention

Learn to identify and mitigate potential defects early in the development lifecycle, significantly reducing downstream costs and improving overall software reliability.



Agile Optimization

Integrate defect prediction seamlessly into your agile workflows, enabling faster iterations while maintaining the highest quality standards.



Core Methodology: Risk to Defects

Assess Program Risk

Calculate defect density by analyzing historical data and current project characteristics and factors.



Multiply by Scope

Apply defect density to project scope size derived from story points and effort estimates.



Forecast Arrival Times

Predict when defects will be discovered based on installed sites and deployment patterns.



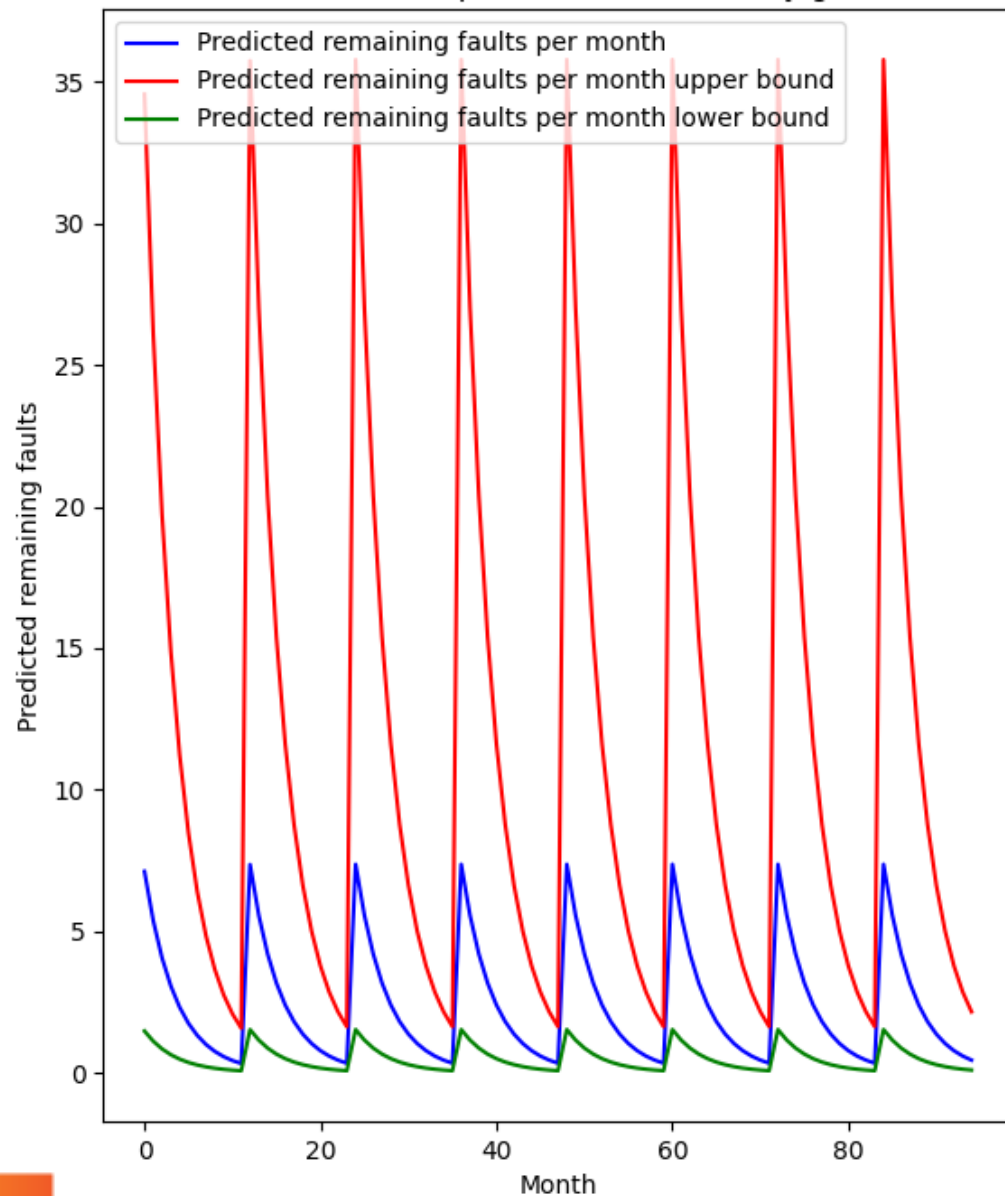
Generate Predictions

Create concrete defect counts and arrival curves for proactive resource allocation and planning.

Factors Affecting Software Defects

Research-proven factors that mathematically correlate with escaped defects, based on USAF Rome Laboratories' foundational 1987 prediction model and continuous refinement since 1993.

Factor Type	Characteristics (%)	Impact Level	Assessment Tools	Examples
Product	50 (10%)	Medium	Static Analysis	Size, complexity, OO design
Process	121 (23%)	High	SEI CMMi, ASPICE	Procedures, standards, compliance
Technique	302 (58%)	Critical	MBSE Tools	Methods, approaches, tools
People	38 (7%)	Medium	HR Metrics	Turnover, experience, team size
Product Risks	12 (2%)	Low	Risk Assessment	User risks, regulations

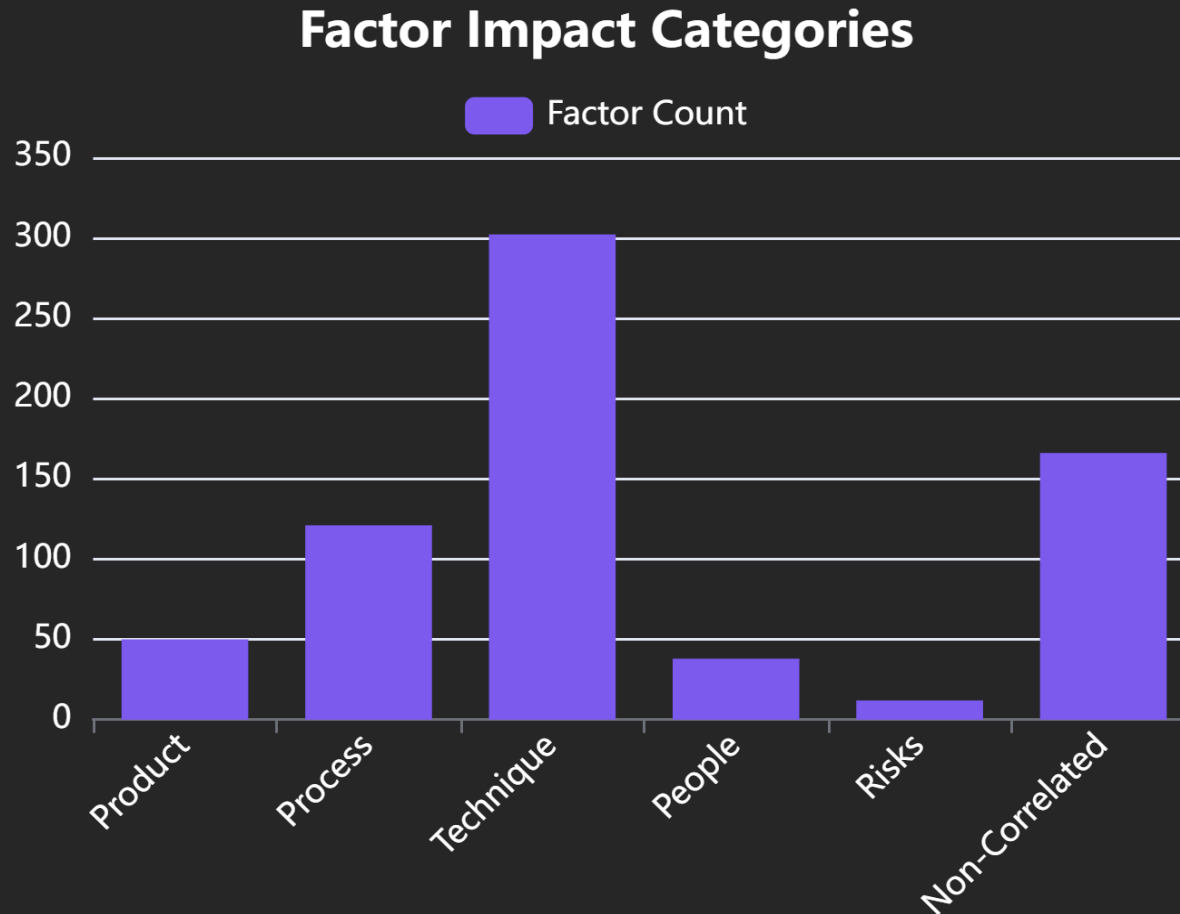


Technical debt is always a risk for both Agile and Waterfall development.

Learn how to forecast defect arrival times in an agile environment.

Defect arrival times
and technical debt

Defect Factor Distribution Analysis



Technique Dominance

58% of all factors relate to specific development methods, tools and approaches used.



Often Overlooked

166 factors don't correlate with defects yet steal resources from factors that matter.

Quantitative Metrics for Project Control

Early Defect Prediction

Predictive Analytics

Learn to quantitatively predict total testing and operational defects with high accuracy and precision.

- Total defects across testing and operations phases.
- Defect discovery rates and timing patterns.
- Technical debt accumulation and impact metrics.

75%+ Defect discovery

Risk Translation

Transform abstract risk predictions into concrete, actionable defect counts and arrival curves.

- Convert risk assessments to defect numbers.
- Generate accurate arrival time forecasts.
- Create actionable project control metrics.

Improved Staffing Accuracy

Resource Planning

Accurately forecast warranty maintenance effort and required staffing levels for optimal resource allocation.

- Warranty maintenance effort calculations.
- Optimal staffing level predictions.
- Proactive resource allocation strategies.

Technical Debt and Defect Timing

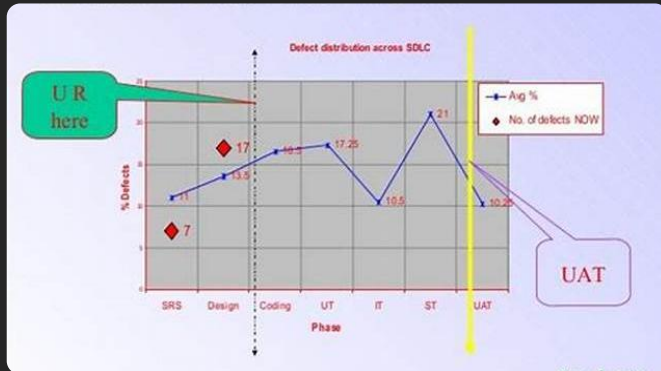
Universal Risk Factor

Technical debt represents a constant risk for both Agile and Waterfall development methodologies. Understanding its impact patterns is crucial for maintaining software quality and predicting future defect scenarios in any development environment.

Agile Forecasting Methods

Master specialized techniques for forecasting defect arrival times specifically within agile environments. Learn to account for sprint cycles, continuous integration, and iterative delivery patterns when predicting defect emergence and resolution timelines.

Who Should Attend This Course



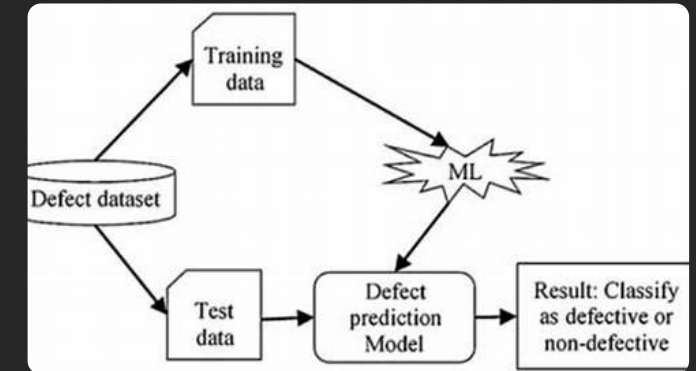
Software Test Engineers

- Predict defects before testing begins for preparation.
- Optimize test strategies based on prediction data.
- Improve resource allocation and test planning efficiency.



RAM Site Reliability

- Essential foundation for advanced failure rate prediction.
- Learn MTBF and availability forecasting methods.
- Enhance site reliability engineering capabilities significantly.



Software Management Teams

- Predict escaped defects impacting future increments.
- Enable proactive resource allocation and planning.
- Prevent defects from wreaking havoc downstream.

Course Delivery Options and Contact

Flexible Learning Formats

Choose from virtual self-guided learning for independent study or virtual instructor-guided sessions for interactive learning experiences. Both formats provide comprehensive coverage of defect prediction methodologies tailored to your team's needs.



Self-Guided Virtual

Complete the course at your own pace with comprehensive materials, practical exercises, and real-world case studies available online.



Instructor-Guided Virtual

Join live virtual sessions with expert instructors for interactive learning, Q&A sessions, and collaborative problem-solving experiences.



Schedule Consultation

Contact
sales@missionreadysoftware.com today to discuss your specific needs and customize the perfect learning solution.



MISSION READY SOFTWARE

Ready to Transform Your Development Process?

Contact sales@missionreadysoftware.com today to schedule your consultation
and start optimizing your software development trade-offs.

