Biography

Josh VerHage is an engineering professional with over 10 years experience in product design. He earned his Bachelor of Science in Electrical Engineering from Michigan State University, and more recently his Masters Degree in Product Design Engineering from Grand Valley State University. His engineering experience includes product design relating to transformers and power supplies, construction heating equipment, and floor cleaning equipment. He is currently working as a Test Engineer with Tennant Company, a worldwide leader in design and manufacture of cleaning equipment.

A Quantitative Approach to Application Testing

Presenter: Josh VerHage, PE, PMP

Co-Author: Dr. Shabbir Choudhuri

Agenda

- Overview of Tennant Company, products
- Product development toolbox
- Flaws of application testing
- Application testing:
  - Quantification
  - Pass/fail determination
  - Assigning value (cost/time)
- Conclusion / Questions

Tennant Company

"Designing, manufacturing, and marketing solutions that help create a cleaner, safer, and healthier world"

- Est. 1870
- Global presence
- Industry leading warranty

How They are Really Used

- Dirty environments
- Unintended use
- Reasonable misuse
- Rental machines
- Untrained operators
- Questionable maintenance

➢ **Can you always model and predict product use?**

➢ **How can we quantify what is unpredictable?**

Product Development Toolbox
Design – Build – Test

Best Practices

- Design Failure Mode and Effects Analysis (dFMEA)
- Accelerated testing
- Reliability testing
- Design of Experiments (DOE)
- Quality/reliability

Practical

- HALT
- FMVT
- Field/customer testing
- Foreseeable misuse
- **Application testing**

Product Application Testing

- This can take many forms:
  - Testing of systems (or subsystems) in a simulated application
  - Automated testing of equipment in a simulated application
  - Prototypes tested by select customers
  - Supervised testing at select application sites
  - **Internal testing in a simulated "real world" environment**

Product Application Testing

We do it to:

- Simulate customer use of the product
- Better understand the product from a customer perspective
- Identify unexpected or unforeseen product failure modes
- To build confidence that the product is ready for the customer

Unanswered Critical Questions

- How long should we conduct testing?
- How much resources to invest?
- What are the pass/fail criteria? Do they exist?
- How do we measure success?
- How can we relate the cost investment to organizational/project success?

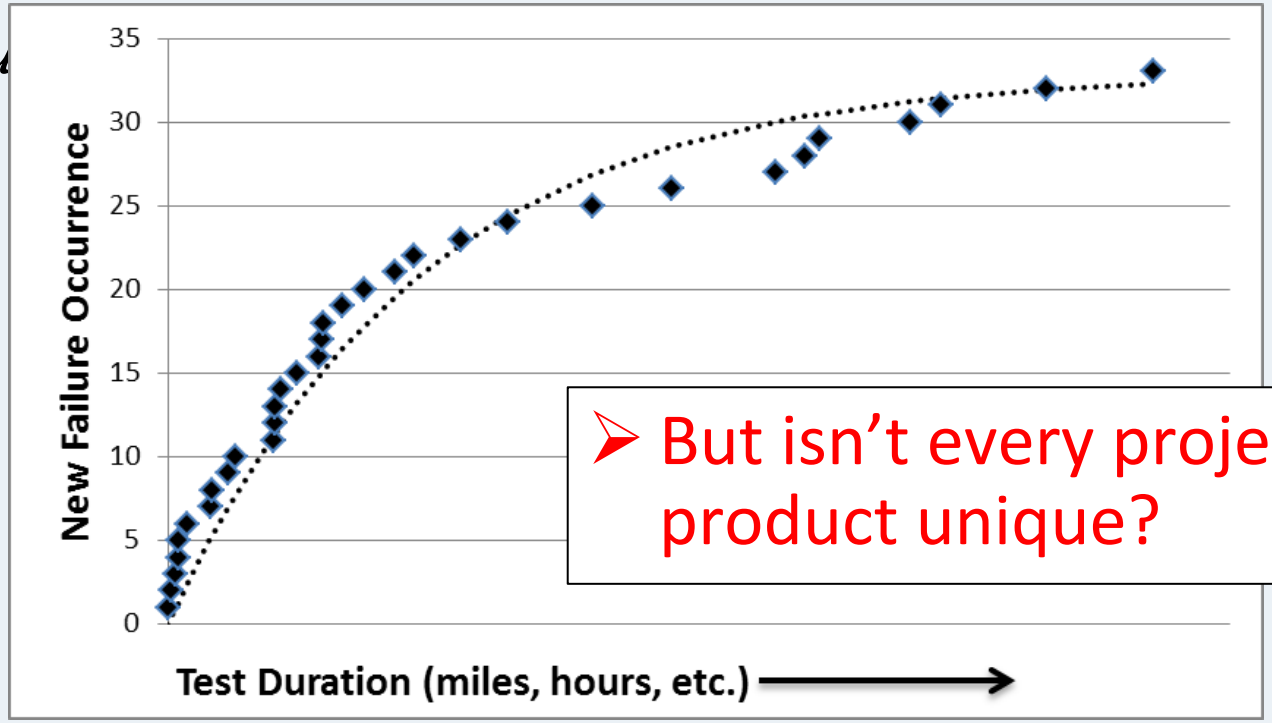
➢ This is often determined through experience and professional judgment.

How to Quantify the Practical Approach?

1. Need a measurable result: Identify new failure modes
2. When is testing "complete" (pass/fail): Based on rate of new failure mode identification
3. Relate application testing to product success: Associate failures with a direct cost to the organization

Application Test Hypothesis

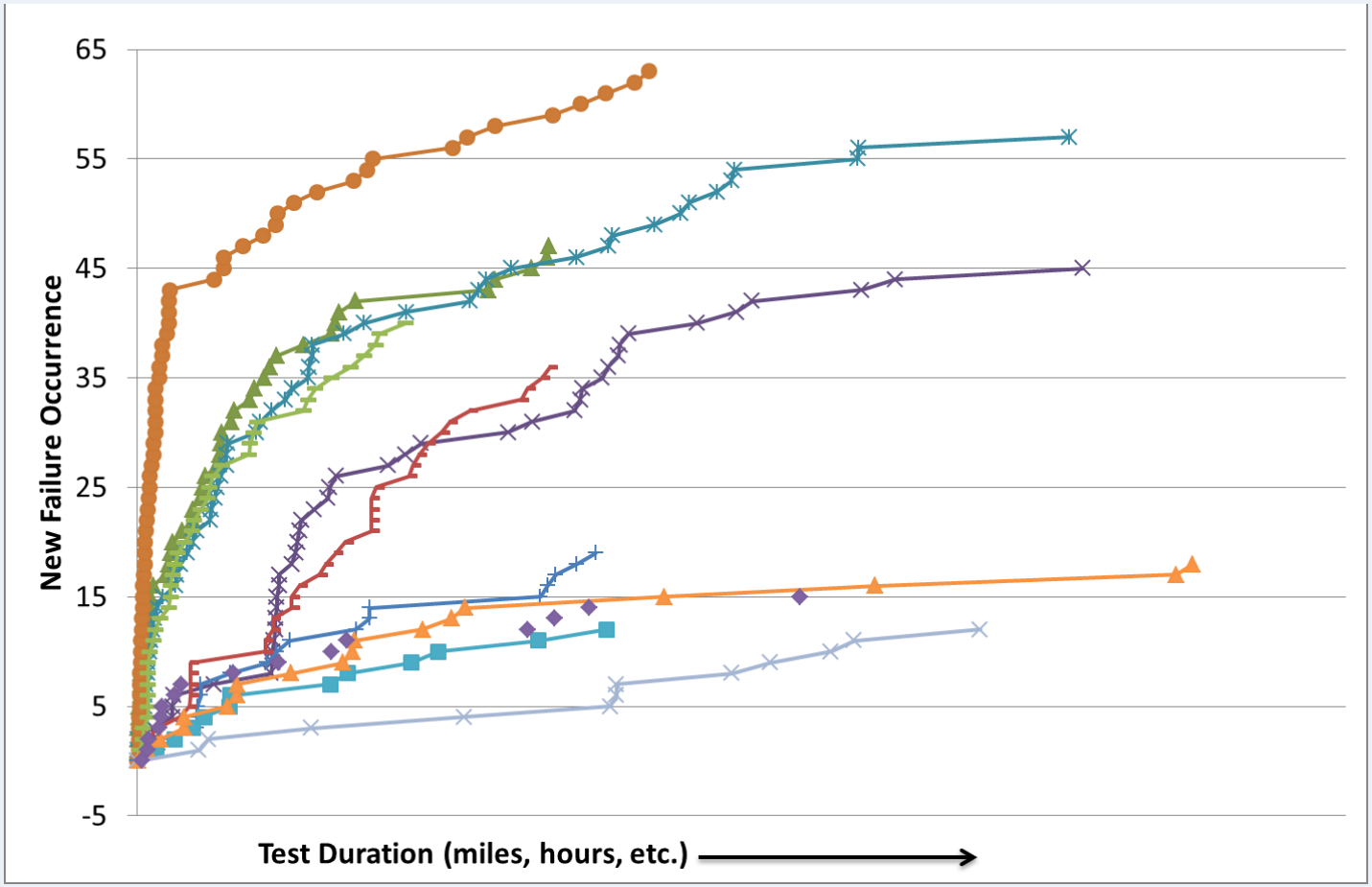
- New failure identification rate will decrease over time
- Decrease in learnings represents decreased test value
- Model the Data: $\# Failures = \theta_1 - \theta_1 e^{(-\theta_2 * hou}$



➢ But isn't every project and product unique?

Historical Project Evaluation

- Leverage history and experience (11 projects):



Historical Project Evaluation

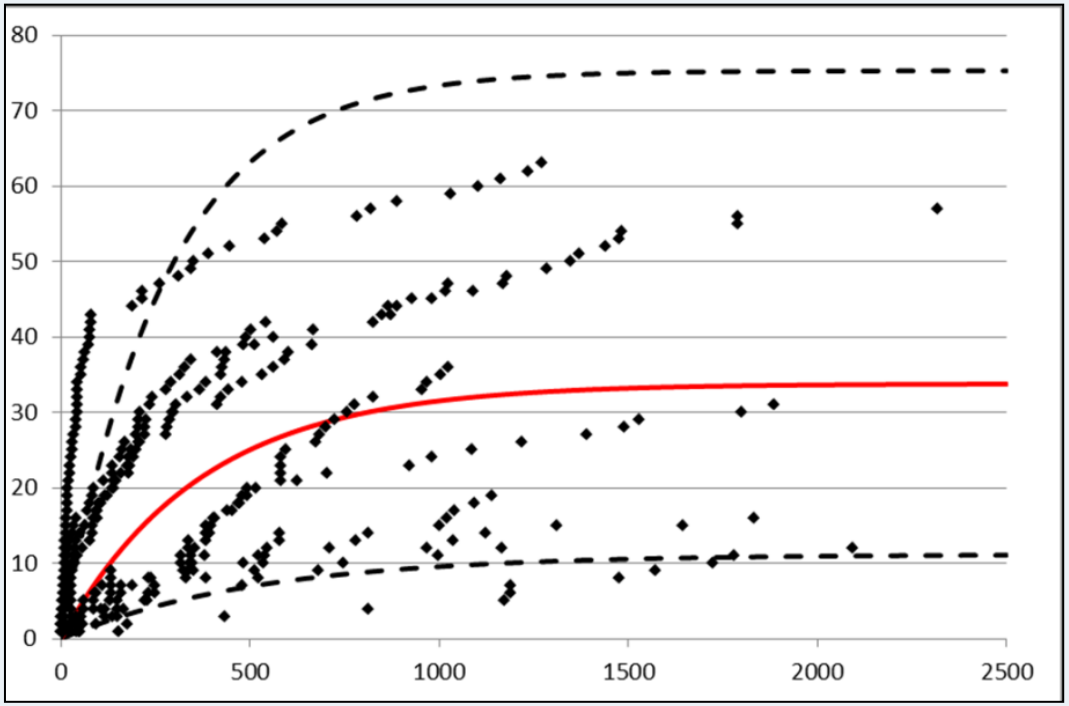
- Projects variables:
  - Intended application
  - Product purpose
  - Retail price
  - Sales volume
- Quantity of identified failure modes varied
- Projects appear to exhibit consistent trends

Other Data Uses (Suggestions)

- Evaluate project factors: How do they correlate to failures?
  - Machine type (complexity)
  - Application
  - Market
  - Test setup
  - Test documentation
- Split failures by product subsystem

A Regression Model

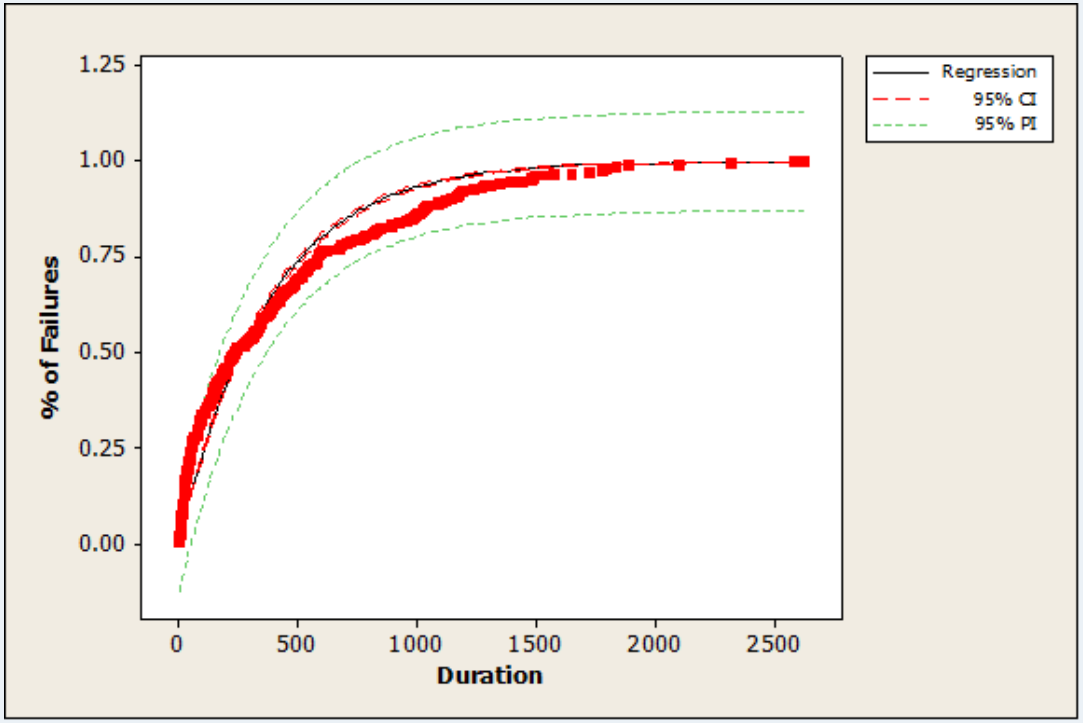
- Fit data to a concave regression curve
- LARGE 95% confidence interval
- But this provides our measurable response!



$$=\theta\downarrow 1 - \theta\downarrow 1\ e\uparrow(-\theta\downarrow 2 * hours)$$

A Regression Model (Continued)

- Pool the project data, and normalize
- Measure value based on % failures identified
- So we can now set a pass/fail criteria:
  - How far up the curve to go?



Practical Significance of Unidentified Failures

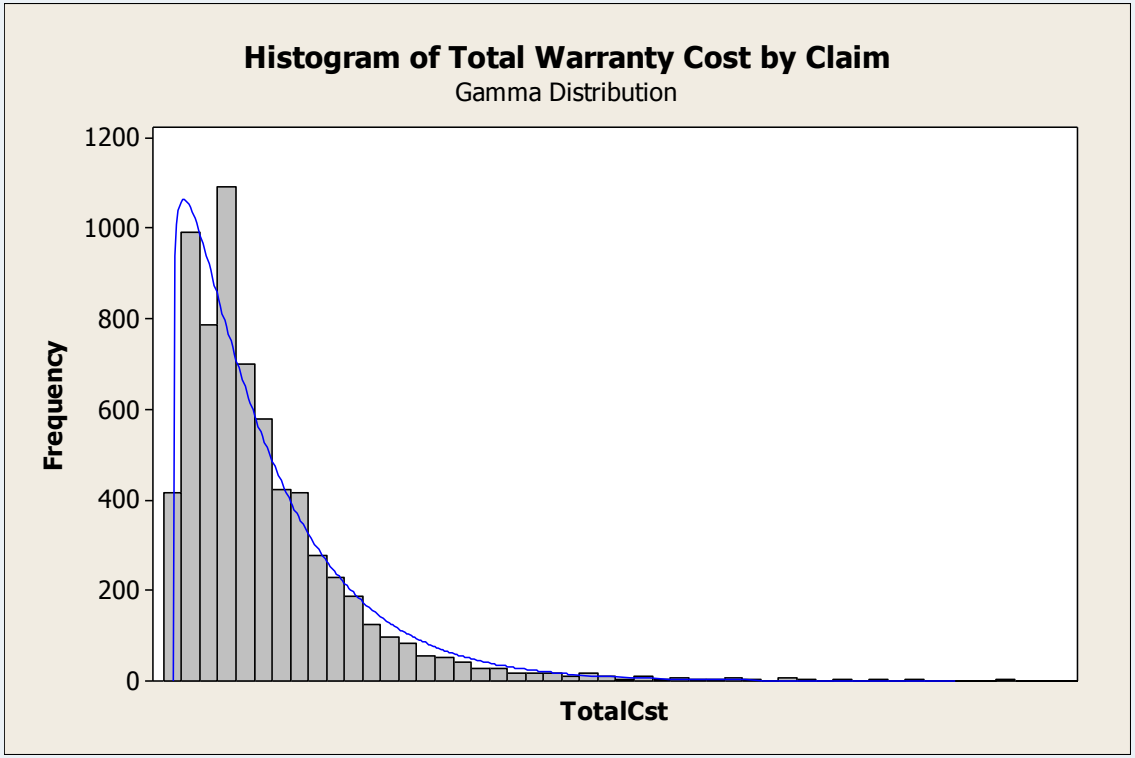
- An unidentified failure = a potential customer concern, or warranty claim
- Factors that contribute to increased project risk are:
  - High warranty costs
  - High sales volume (product exposure)
  - Customer perception risk
  - Safety concerns

Historical Warranty Evaluation

- Provides financial context to unidentified failures
- Understand the mean cost of a warranty failure
- Evaluate the distribution of warranty claim cost
- Understand how warranty claim costs are distributed, and what factors influence them:
  - Machine
  - Component
  - Subsystem
  - Etc.

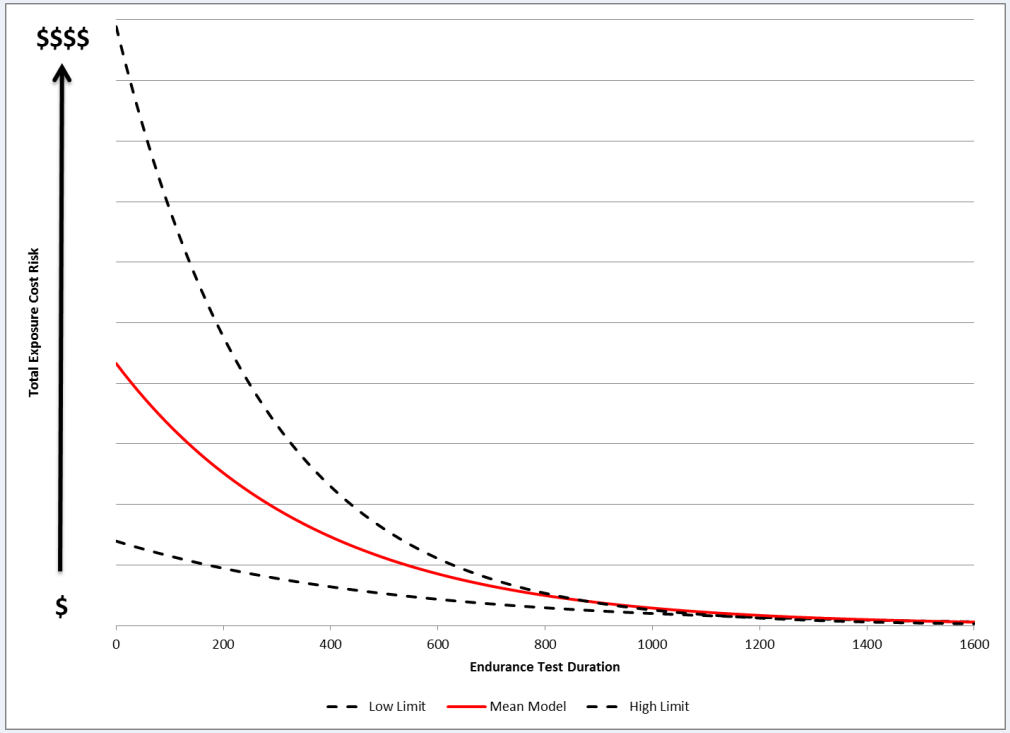
Warranty Data: An Example

- Evaluation of Tennant warranty claim costs
- Realized a relatively predictable Gamma distribution
- Reasonable to predict an average and variation of warranty claim cost



Combine Test and Warranty Data

- Combine:
  - Diminishing failure rate
  - Warranty claim cost
  - Projected sales
- Notice risk reduced and more predictable as testing progresses



$$Potential\ Exposure\ Cost = Cn\downarrow a\ \theta\downarrow 1\ e\uparrow(-\theta\downarrow 2 * duration)$$

C = Mean Claim Cost

$n\downarrow a$ = Number of Failures Predicted

$\theta\downarrow 1$, $\theta\downarrow 2$ = Constants

Conclusions

- Application testing: A great way to understand a noisy product environment
- Follow the three steps to build your model:
  1. Need a measurable result: Identify new failure modes
  2. Determine a pass/fail criteria: Diminishing rate of failures
  3. Relate application testing to product success: Associate failures with a direct cost to the organization
- Be creative: Use the data you have to better understand your own products

Questions/Comments