Technical Software | Electronics | Industrial Mathematics | Remote Solutions

SIOUX
SOURCE OF YOUR TECHNOLOGY

KEY CONSULT
PROCESS INNOVATION

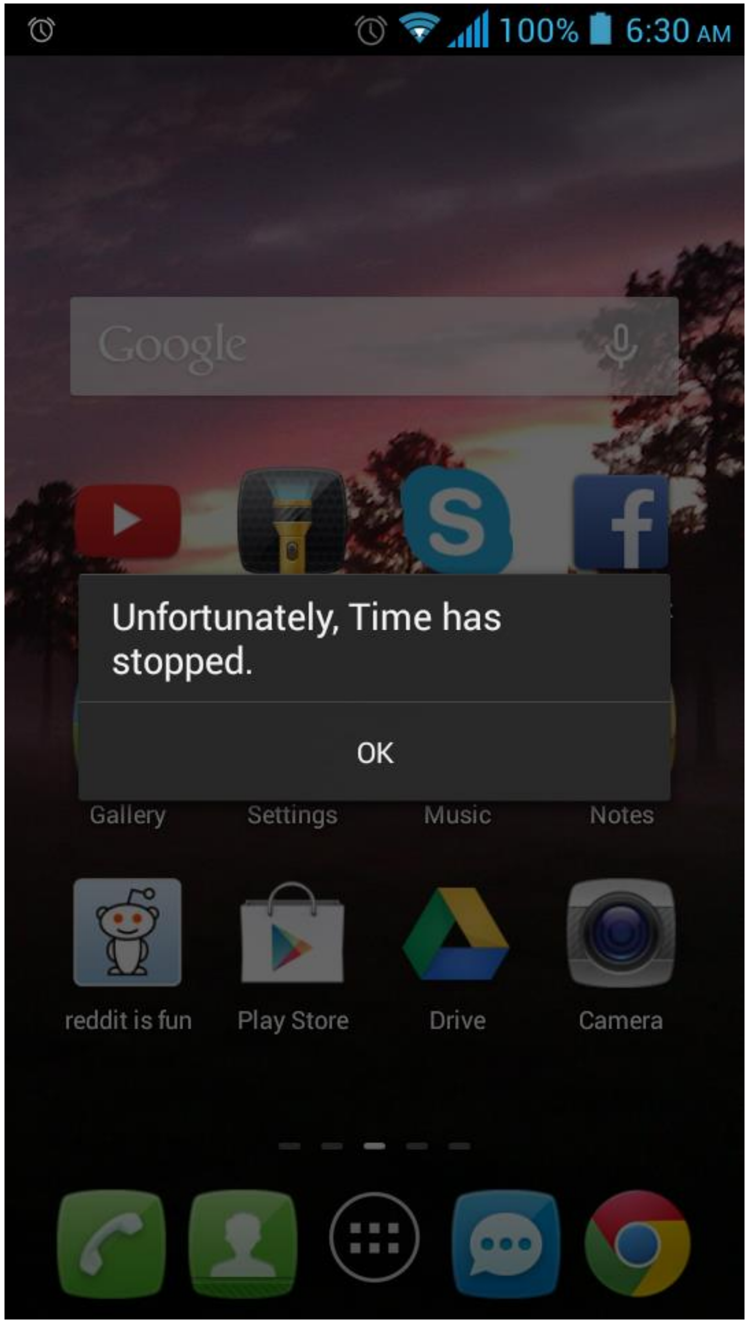
Finally… Reliable Software!

21E NEDERLANDSE TESTDAG

Bryan Bakker
Rob de Bie

Eindhoven 2015

bryan.bakker@sioux.eu rob.debie@key-consult.nl
@Bryan_Bakker



Unfortunately, Time has stopped.

OK

Technical Software | Electronics | Industrial Mathematics | Remote Solutions

SIOUX
SOURCE OF YOUR TECHNOLOGY

KEY CONSULT
PROCESS INNOVATION

Finally… Reliable Software!

21E NEDERLANDSE TESTDAG

Bryan Bakker
Rob de Bie

Eindhoven 2015

bryan.bakker@sioux.eu rob.debie@key-consult.nl
@Bryan_Bakker

Contents

- Intro
- Software reliability
- Four step model
- Different steps
- Conclusion

SOURCE OF YOUR TECHNOLOGY

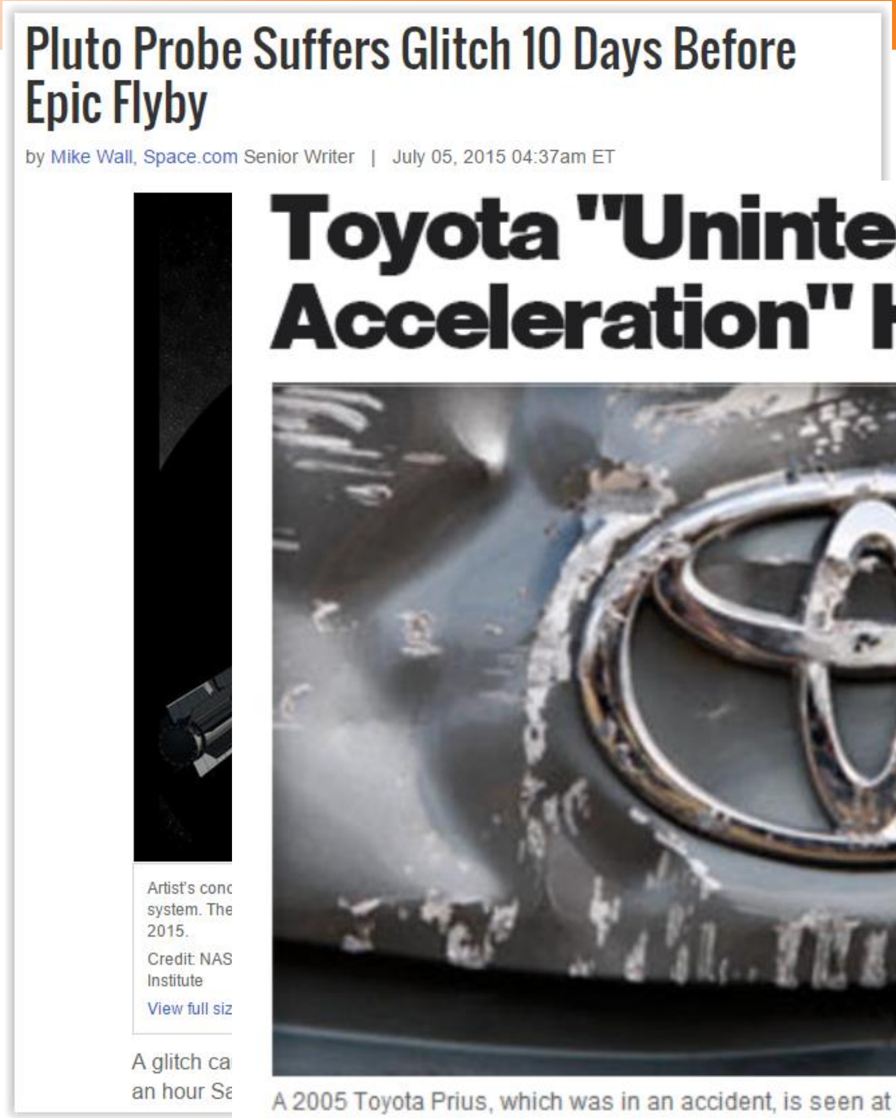
PROCESS INNOVATION

- Test Architect
- Certifications: ISTQB, TMap, Prince2
- Member of ISTQB Expert Level on Test Automation
- Tutor of several test related courses
- Domains: medical systems, professional security systems, semicon-industry, electron microscopy
- Specialties: test automation, integration testing, design for testability, reliability testing

- Consultant
- Owner of Key Consult
- Development process definition, assessment and improvement support
- Royal Philips, NXP Semiconductors, ASML, Texas Instruments, Sensata
- Domains: Medical equipment, consumer electronics, semiconductor industry
- Europe, USA, Asia
- PMBoK, CMMI, SCRUM

SIOUX
SOURCE OF YOUR TECHNOLOGY
KEY CONSULT
PROCESS INNOVATION

Reliability


Pluto Probe Suffers Glitch 10 Days Before Epic Flyby

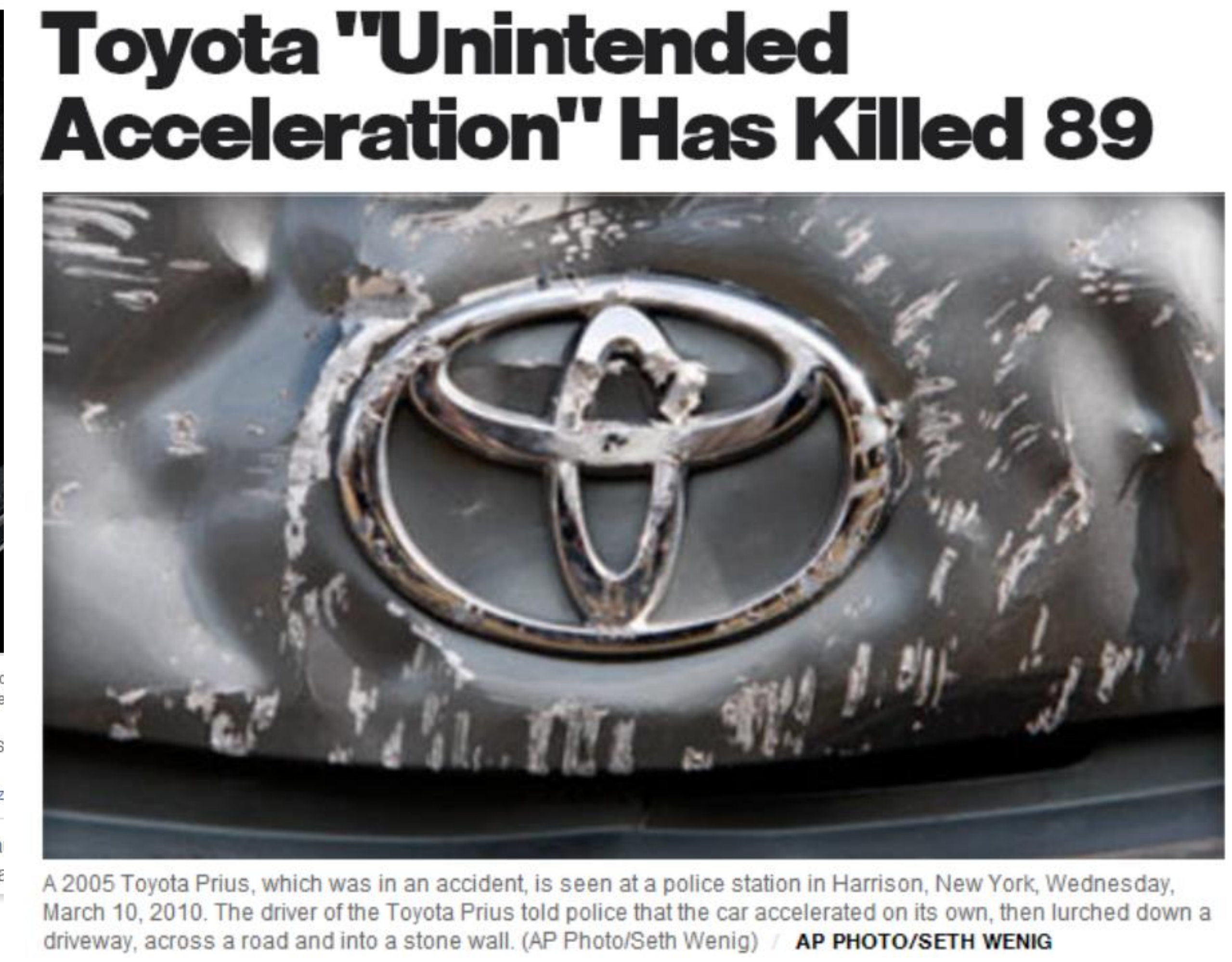
by Mike Wall, Space.com Senior Writer | July 05, 2015 04:37am ET

Artist's conc
system. The
2015.

Credit: NAS
Institute

View full siz

A glitch ca
an hour Sa


Toyota "Unintended Acceleration" Has Killed 89

A 2005 Toyota Prius, which was in an accident, is seen at a police station in Harrison, New York, Wednesday, March 10, 2010. The driver of the Toyota Prius told police that the car accelerated on its own, then lurched down a driveway, across a road and into a stone wall. (AP Photo/Seth Wenig) / **AP PHOTO/SETH WENIG**

SIOUX
SOURCE OF YOUR TECHNOLOGY
KEY CONSULT
PROCESS INNOVATION

Software Reliability

- What is software reliability?

“Software Reliability is the probability of failure-free software operation in a specified environment for a specified period of time.”
IEEE 729

In short:

- Something can be functionally correct
- But is it reliable? How reliable is it?

SOURCE OF YOUR TECHNOLOGY

PROCESS INNOVATION

Four steps

Define user domain reliability targets

1

Derive software reliability targets

2

Define engineering processes

3

Measure software reliability growth

4

Example: Security and Surveillance System

- Cameras
- Recording
- Event Handling

First step

Define user domain reliability targets:

- Define customer profiles
- Identify operation modes
- Determine reliability targets per operation mode

SOURCE OF YOUR TECHNOLOGY

PROCESS INNOVATION

First step

- Define customer profiles
  - ATM security

Define user domain reliability targets

1

SIOUX SOURCE OF YOUR TECHNOLOGY KEY CONSULT PROCESS INNOVATION

First step

- Define customer profiles
  - ATM security
  - Parking lot surveillance

Define user domain reliability targets

1

SOURCE OF YOUR TECHNOLOGY

PROCESS INNOVATION

First step

- Define customer profiles
  - ATM security
  - Parking lot surveillance
  - Airport surveillance

Define user domain reliability targets

1

First step

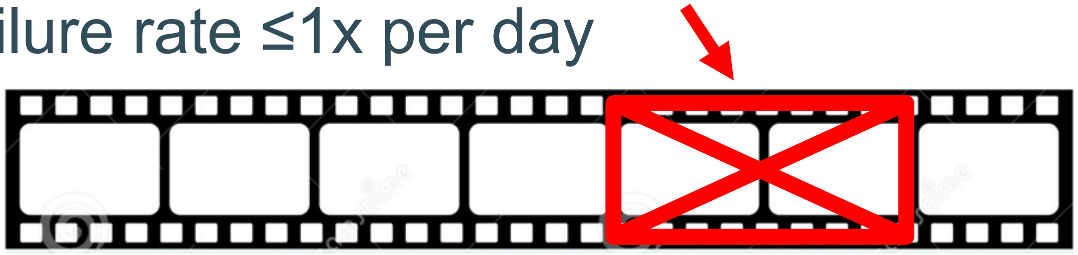
Define user domain reliability targets

1

- Identify operation modes
  - Recording mode
  - Playback mode
  - Auto-start
    - Out-of-the-box
    - User triggered
    - Software reset

First step

- Determine reliability targets per operation mode
  - Segment between 0.5s and 2s missed → failure rate ≤1x per day



  - Playback command does not function as expected → failure rate ≤ 1x per hour of viewing
  - Not auto-started → failure rate ≤ 3 * 10^{-7} failures/restart

Define user domain reliability targets

1

Second step
Operational profile

Derive software reliability targets

2

- How will the product be used by customers?
- Operational profile: a quantitative characterization of how a system will be used
- Developed by John Musa
- Drives reliability engineering, e.g.:
  - Reliability testing
  - Design decisions (e.g. robustness)

SOURCE OF YOUR TECHNOLOGY

PROCESS INNOVATION

Second step

Derive software reliability targets

2

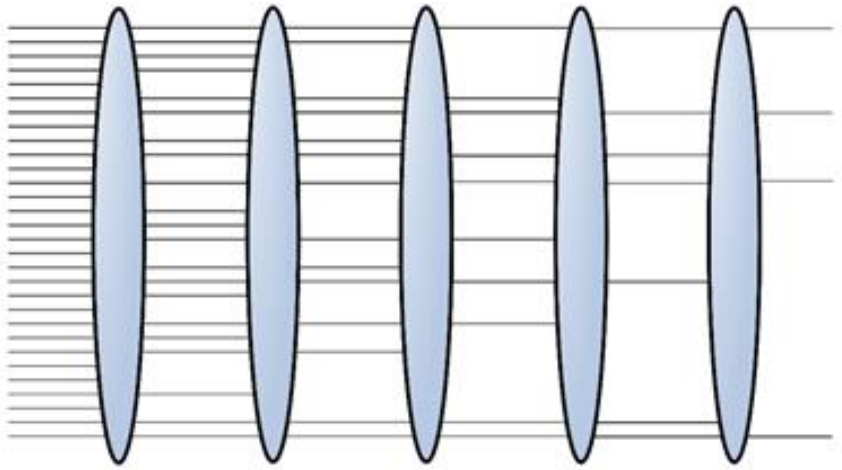
- Define operational profile for user functions

Playback function	Occurrence	Probability %
Play / Pause (toggle)	48 out of 100	48%
Fast forward	20 out of 100	20%
Fast reverse	20 out of 100	20%
Setup playback windows	10 out of 100	10%
Search and select event for playback	2 out of 100	2%

- Decompose software reliability targets
  - Identify contributing software components
  - Determine reliability targets per software component

Third step

- Define the engineering processes
  - Process steps to prevent reliability faults
  - Process steps to detect reliability faults



  - Design choices to minimize effects of faults

Define engineering processes

3

SIOUX
SOURCE OF YOUR TECHNOLOGY
KEY CONSULT
PROCESS INNOVATION

Fourth step

<u>Measure software reliability growth</u>
4

- Measure software reliability growth
  - Design and execute reliability tests, based on the operational profiles
  - Randomly execute test set according to the operation profile
  - By compression of the profile, test execution can be accelerated
  - Visualize reliability growths

SOURCE OF YOUR TECHNOLOGY

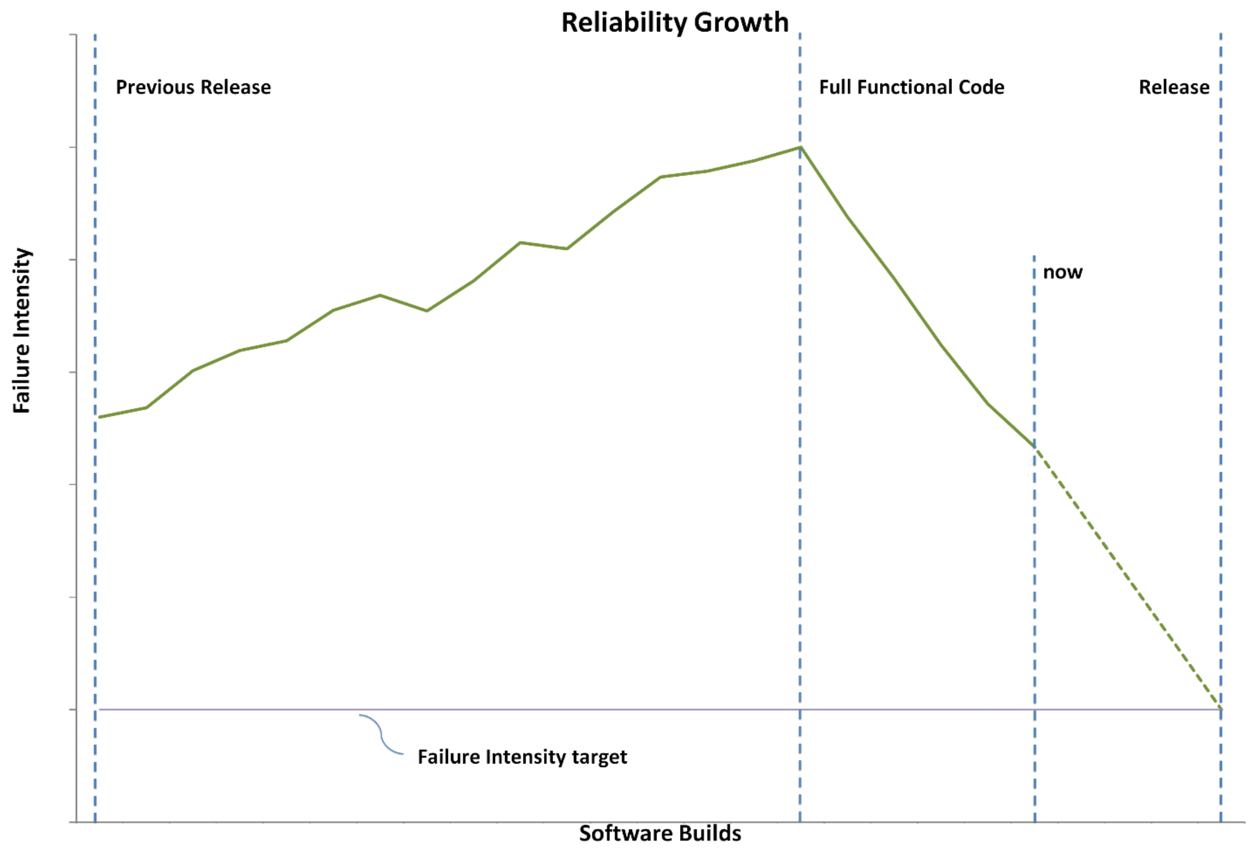
PROCESS INNOVATION

Fourth step

Measure software reliability growth

4

- Reliability growth curve



SIOUX SOURCE OF YOUR TECHNOLOGY KEY CONSULT PROCESS INNOVATION

What about?

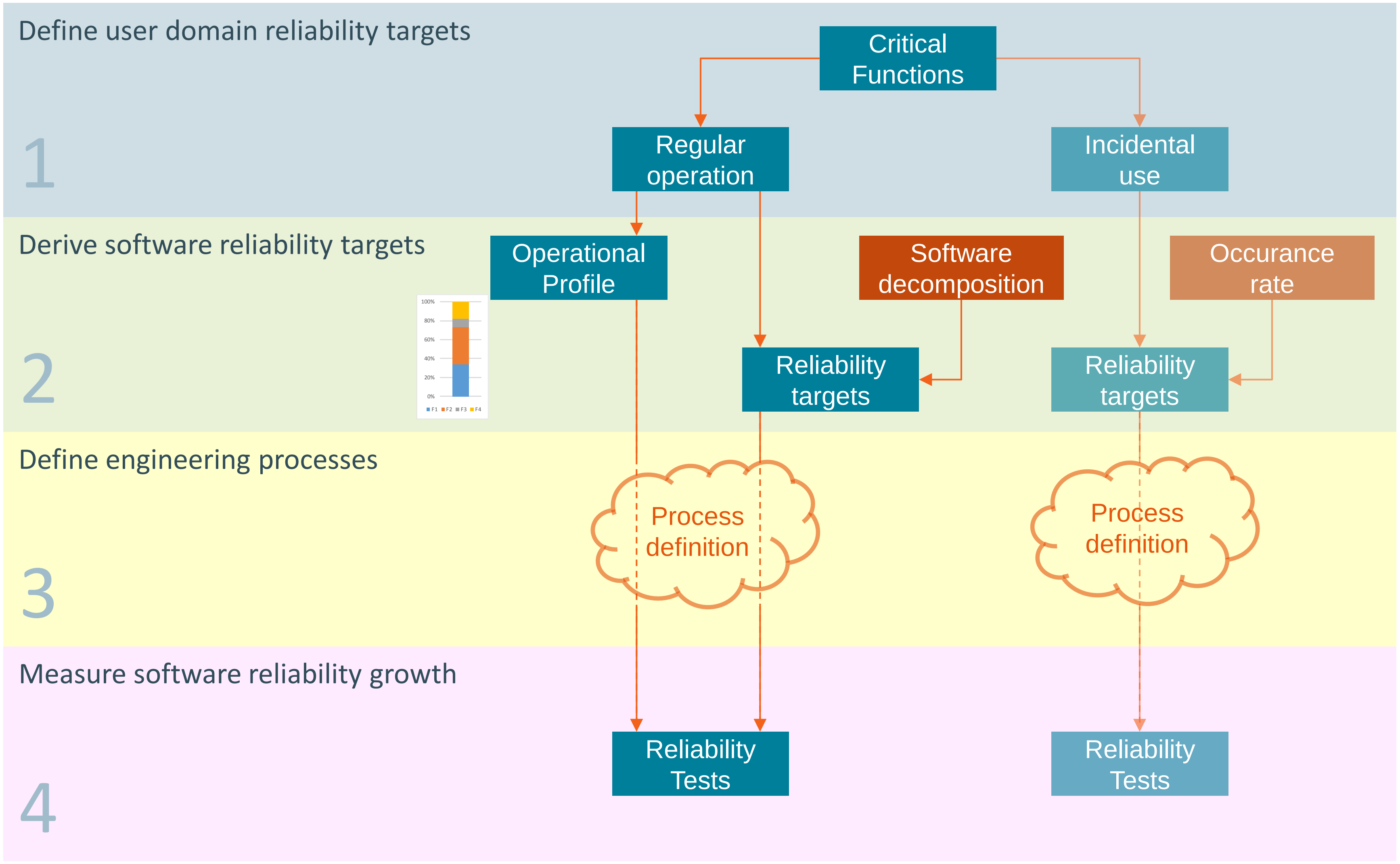
- Critical functions can be missed in operational profiles.
- Treat them separately!

Looking back at first step

- Identify operation modes
  - Recording mode
  - Playback mode
  - Auto-start
    - Out-of-the-box
    - User triggered
    - Software reset

Critical function

Critical functions



SOURCE OF YOUR TECHNOLOGY

PROCESS INNOVATION

Incidental critical functions

- For reliability testing:
  - Different set of tests focused on specific function
  - Higher time compression
  - Separated reliability growth curves for specific functions

Conclusion

- Reliability is not binary but a characteristic that can be measured
- Reliability is not reached by coincidence
- Practical 4 step engineering approach available
  - Based on theory of John Musa
- For a full description and elaborated case study, see next page…

Questions

Published February 2015:

Rob de Bie

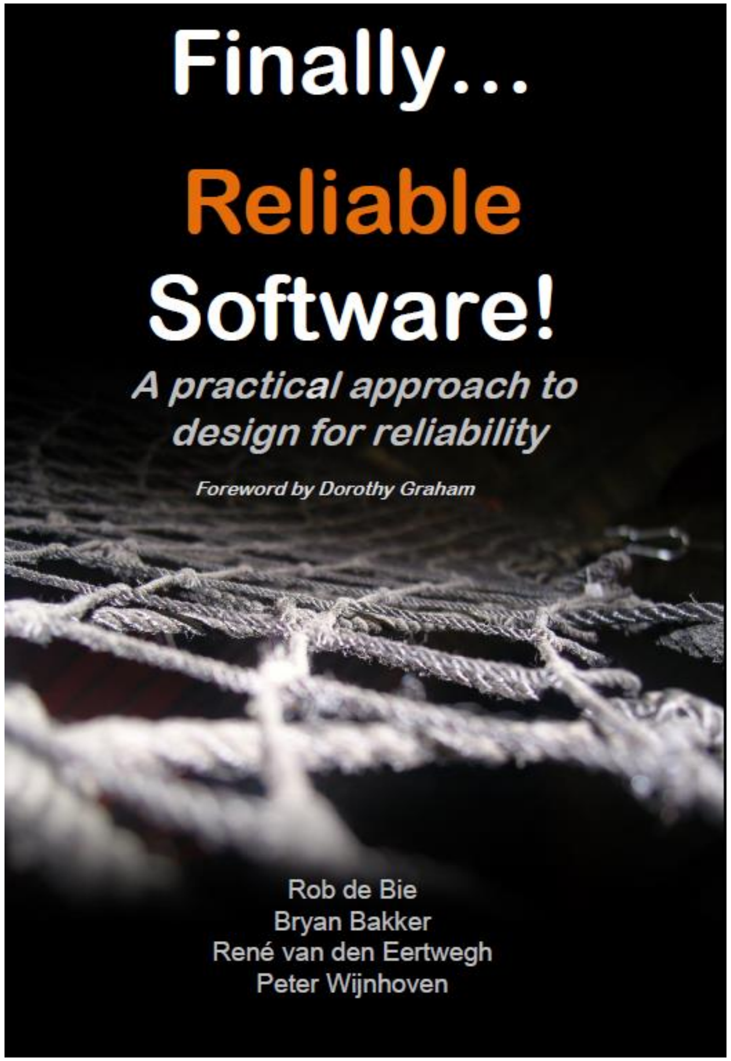
Bryan Bakker

René van den Eertwegh

Peter Wijnhoven

www.amazon.com

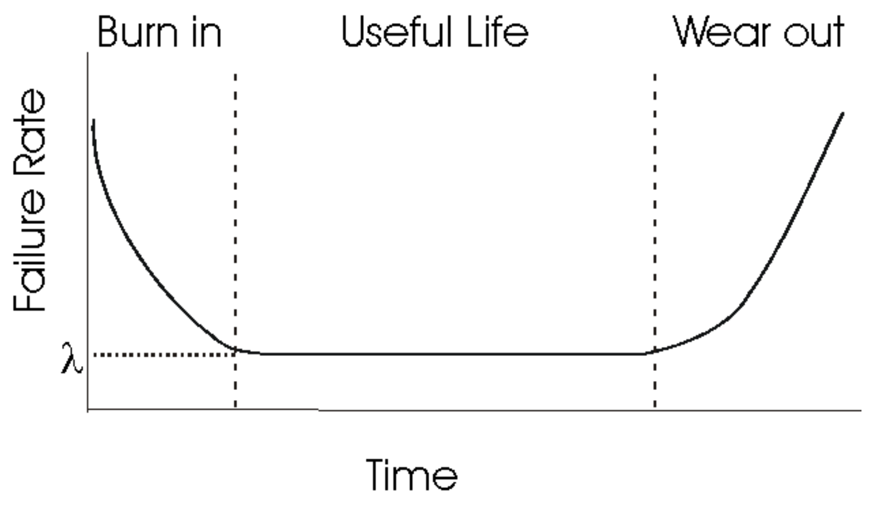
ISBN: 978-1499226669



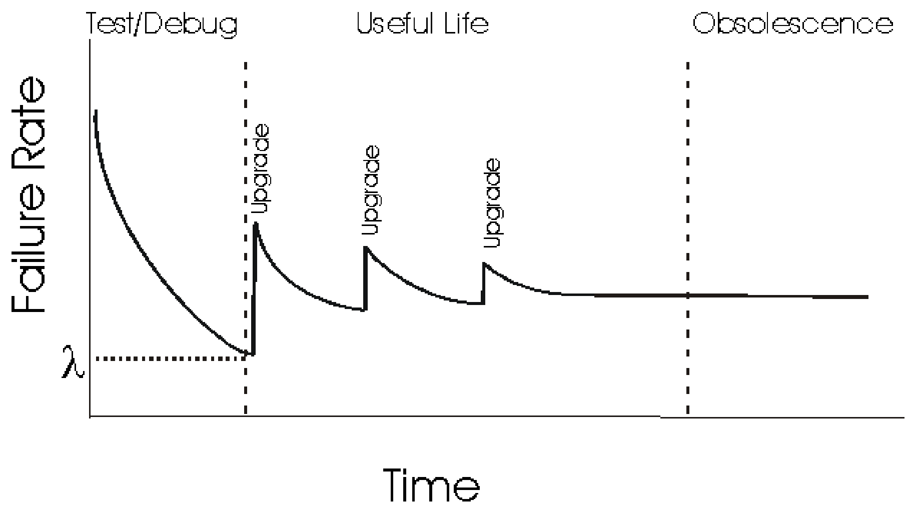
- Backup slides

Reliability Curves

- **Bathtub curve**
  Hardware Reliability



- **Sawtooth curve**
  Software Reliability



SOURCE OF YOUR TECHNOLOGY

www.sioux.eu

+31 (0)40 26 77 100

@ bryan.bakker@sioux.eu