It's a myth that quality costs more. In creating software, quality costs less.

James Over, Opening Remarks

"What would it take today?"

"Back pain. The tie that binds."

TSP helping to lay the foundations of the next generation of software.

"You are here."

TSP 2014 Keynote: Case Study of Toyota Unintended Acceleration and Software Safety

Philip Koopman
Carnegie Mellon University

Code Complexity

McCabe Cyclomatic Complexity Metric

ETCS Code Throttle Angle Function complexity Complexity equals 17

"No configuration management"

"No bug tracking system"

Global Variables are evil.

ETCS -> 19,000 Global Variables

Concurrency Bugs Race Conditions

ETCS Recursion

Recursion can lead to pushing copies of data into stack, leading to stack overflow.

Toyota ETCS

Main CPU

Main chip

Fault containment region

Fault containment region

Single Point of Failure

Monolithic Race

Concurrent copies of VPs go through the same input data on the same chip.

Update from Task 1 gets overwritten by Task 2.

Concurrency and race conditions cause data loss.

Global state does not persist across stack frames.
The Rest of the Story: From Product Process to Process Analysis
Jeff Schwalb, NAVAIR

Learn a little, do a little.

- TEL is based on the application of good engineering in all technical areas, not just software.
- Defining a team’s process supports effective planning during launch.
- Effective post mortem analysis relies on well-defined workflows.

Transitioning with Process Modeling

ENTRY TEST VERIFICATION EXIT (ETVX)

Before

After

Post Mortem Analysis
- Considers artifacts applied during the project
- Looks at data collected throughout project
- Individual Experiences
- Data mining through automated process tool
TSP 2014 Keynote: Wild, Wild West—How to Corral All Your Developers into Creating Secure Code
Jonathan Beck
Vice President and Director of Security Controls, PNC Financial Services Group

Wild, Wild West -
How to corral all your developers into creating secure code.
—Jonathan Beck
PNC Financial Services Group

APPLICATION SECURITY TOUCHES
- Understand how each contributes to making secure software.

Original State
- Static & Dynamic Teams find vulnerabilities.
- Penetration Test impact applications.
- Provided general security requirements.
- Development team fixed bugs.

- DEFECTS DISCOVERED AT MOST EXTREME STAGE OF DEV CYCLE?
- Application Security Coaches
  - Understand our developer's thinking.
  - Cargo in many languages, in many countries.
  - Supporting 100s of apps & financial environments.
  - NO CENTRALIZATION.
  - NO HEAD.

RISK ANALYSIS
- DRIVES SECURITY CONTROLS
- VERY PERSONAL TO AN ORGANIZATION
- RISC FRAMEWORKS
  - Risk
  - Blend

THREAT RISK MODELING
- Looks at RUE FROM THE Vantage POINT OF THE BAD GUY.
- NOT JUST SOFTWARE, BUT OPERATIONAL & BUSINESS PRACTICES
- TRUST Boundaries
- CHALLENGING FOR COMPLEX SYSTEMS
- EXPERTISE needed to lead the discussion.

CAMPAIGN INCLUDED
- Forums
- Awareness Articles
- Security Support
- Presentations in Security Management
- Executive Support
- TRAINING PROGRAM
  - customised for each role
  - monitoring & optimized
  - CBT & instructor led.

Best Practice is the start of your journey, not the end.
TSP 2014: A Zero-Depth Entry to Using TSP: How TSP Turned Around the Smart Grid Maturity Model Project
Summer Fowler, Carnegie Mellon Software Engineering Institute
Julia Mullaney, Carnegie Mellon Software Engineering Institute

A Zero-Depth Entry to Using the TSP: How TSP Turned Around the Smart Grid Maturity Model Project

A Love Story: TSP & Project Management

TSP is Not Just for Software

The team was dispersed, divided their time, lots of turnover
Used launchees & post-mortems as team-building activities

Load Balancing
- Make sure tasks matched % committed

Budget Analysis
- Analyzed data from three approaches to finalize the plan
- Budget Quarterly
- Reiter Weekly
- Reconcile Monthly

Design Lessons Learned
- How to design TSP process to the products produced
- Define usage, audience

Helped Design Navigator Course, Training Course

TSP Setup Team for Easy Implementation

Overall Lessons Learned
- Need better methods to conduct requirements analysis
- Didn’t gather usable historical data
- Stickiness: Experience didn’t transfer to other projects
- Quality was a “journey”
- Dramatic Increase in Value

LAUNCH LESSONS LEARNED
- Planning made Project & Project Team Successful
- Work got done in spite of overcommitment
- Insight into cost => better decisions

And they lived happily ever after
TSP 2014 Keynote: Information Flow: The Secret to Successful Teamwork
Jesse Schell
CEO, Schell Games
TSP 2014: Insider Threats in the Software Development Life Cycle
Daniel Costa, Carnegie Mellon Software Engineering Institute
Randall Trzeciak, Carnegie Mellon Software Engineering Institute

INSIDER THREATS IN THE SOFTWARE DEVELOPMENT LIFECYCLE

"Insider used to mean 'trusted.' Now corruption is negative."

THERE IS NO MAGIC BULLET FOR INSIDER THREATS—HOW DO YOU DETERMINE MALICIOUS INTENT?

PHASES OF SOFTWARE LIFECYCLE:
- Requirements Definition
- System Design
- System Implementation
- System Deployment
- System Maintenance

WHAT'S MOST IMPORTANT TO YOU TO PROTECT?

WHO
WHAT
HOW

"Not everyone is a threat to everything."

INSIDER THREATS INCLUDE:
- Malicious & Unintentional Damage:
- May Even Target Take Sensitive Information with Them When They Change Jobs.

INFORMATION NOT ALL INSIDERS ARE MALICIOUS... INSIDE THREAT IN SPACE...
TSP 2014: Under N: Acceptance to Delivery in N Hours

Umashankar Velusamy
Verizon Communications, Inc.

UNDER N:
Acceptance to Delivery in N hours

Umashankar Velusamy

Under-N

Smart Delivery
of pre-defined
orchestrated software
Changes from
acceptance to delivery
in under "N" hours.

- Driven by business value
- Does not compromising quality
- Co-exist with other business delivery cycles
- Many teams already doing it!

Frame Work:
Create a template with
key info associated with
Under-N capability,
both the info needed &
the info that will be used.

- Ask the right questions
to the IT teams

Governance:

1. Capability Request
2. Define Capability
3. Team Access & Approve
4. Define Template
5. Confirm Task Ownership
6. Ensure Process & standards
7. Launch

Support Structure:

- Strong Executive Support
- Create Under-N Fabric
- Celebrate Success
- Broadcast, Identify & Apply
  Capabilities
- Challenge Teams to Self-Service

When does it fail?

- When used to bypass processes
- When cutting corners
- Wrong choice of tools or Architecture
- When trying to squeeze large projects
When You Build It, They Will Come

Richard Pethia
Director of the CERT Program, Carnegie Mellon Software Engineering Institute

We are dependent on evolving cyber ecosystems...
...and that trend is going to continue.

Cyber security is only going to become more critical.

Software complexity increases vulnerability to attacks.

Open source supply chain is vulnerable.
- Security skills imbalanced among developers.
- No provenance of code.
- No proper fix updates.

More than 91% of development organizations do not coordinate their security practices across the board.

96% of attacks not difficult.
97% of attacks were avoidable.

Design weaknesses drive vulnerabilities.
76% of most dangerous errors stem from design weaknesses.

Static testing and source code analysis improving—leading to better code.

Government systems moving away from check-the-box checks and towards ongoing processes that increase/enhance security resilience.

There is a critical shortage of cyber security professionals.
The need is outpacing supply of trained people.

Organized Crime & Terrorist Networks Involved

The solution:
1. Better software engineering practices
2. Improved security & resilience practices
3. Larger, skilled workforce

How to make secure software agilely and affordably?

Good software architecture leads to longevity & quality in software-reliant systems.

Malware samples accumulating at more than a million a month.

The CERT Program - Carnegie Mellon University

First Stage Adversaries:
- Stealing, using & sharing information.
- Imitating, spoofing & stealing.
- Spreading malware & blocking services.
- Denying damage or denial, extortion.

Second Stage Adversaries:
- Do the real damage, theft, extortion.

Credit Card, Social Security, Intellectual Property

The Perfect Cyber Storm

More & Better Attacker Chain

Growing dependence on interdependent & global systems

Engineering processes to build & maintain secure systems

Cyber crime the biggest threat to our defense, economy, & liberty

All aspects of the internet growth dramatically—
1999: 4 hosts — Today: Over a billion

Neil Potter

The Process Group