



TSP Symposium 2014

Going Beyond Methodology to Maximize Performance

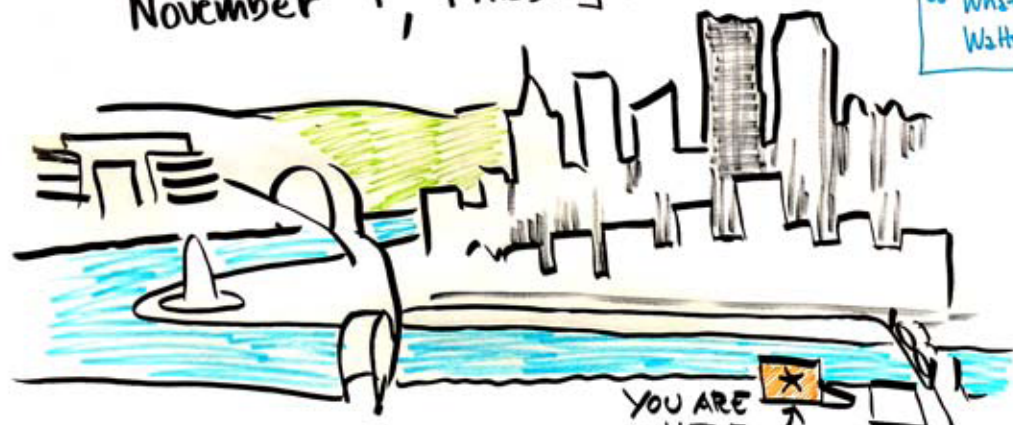
November 3–6, 2014

Pittsburgh, PA

Graphical Recordings

TSP SYMPOSIUM 2014

November 4, Pittsburgh PA



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

James Over, Opening Remarks

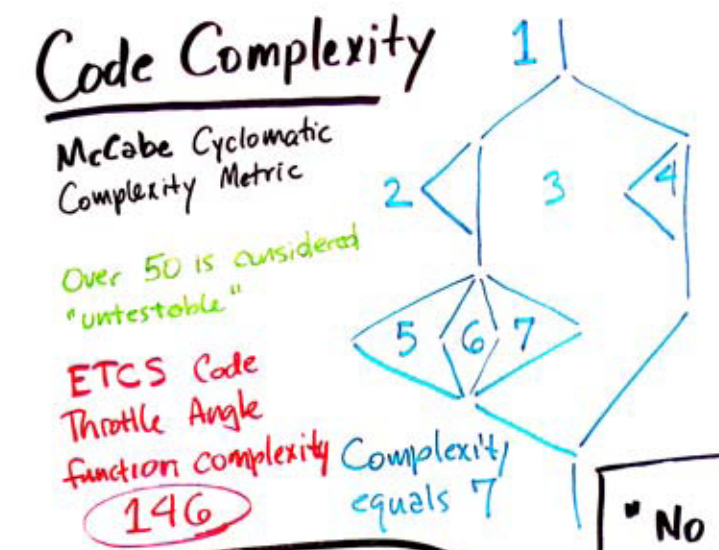
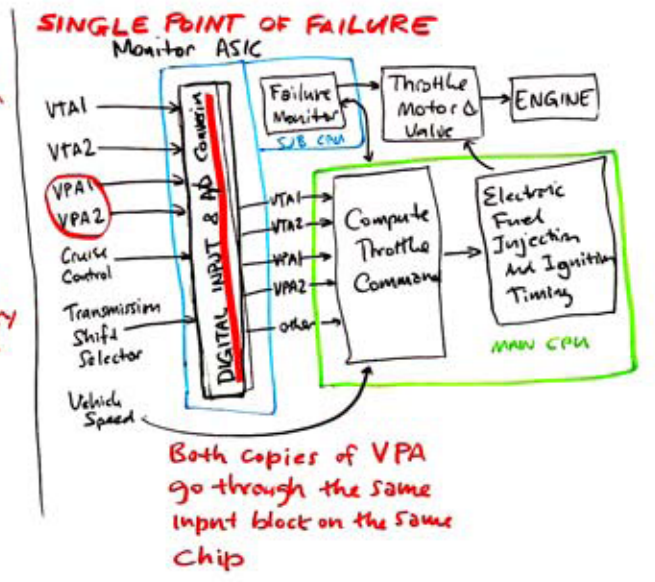
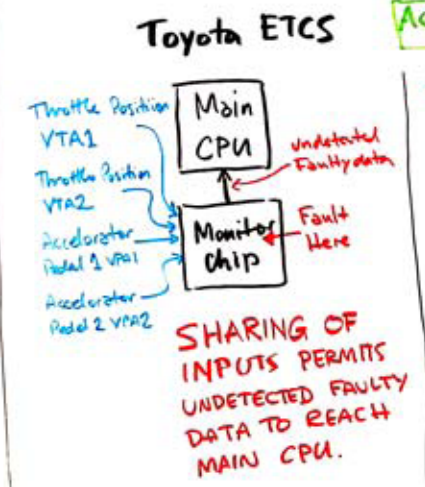
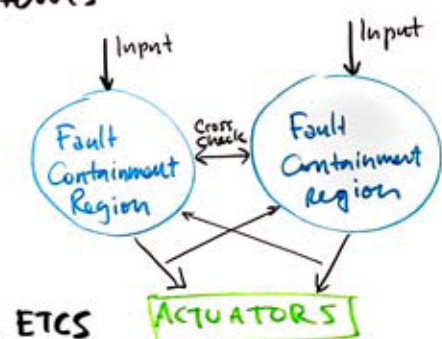
"What would Walt's say?"

"Back pain. The tie that binds"



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

Keynote:
CASE STUDY OF TOYOTA UNINTENDED ACCELERATION AND SOFTWARE SAFETY
- Philip Koopman
Redundancy Required for Critical Systems



Global Variables are evil.

ETCS -> ABOUT 10,000 global Variables

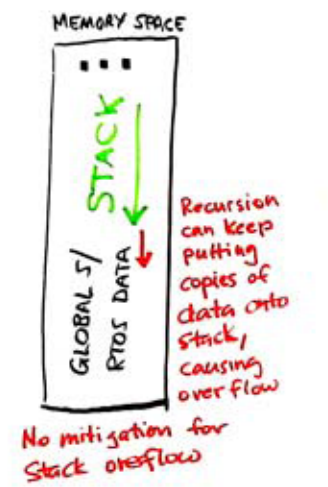
"No configuration management"

"No bug tracking system"

Concurrency Bugs
Race Conditions



ETCS Recursion

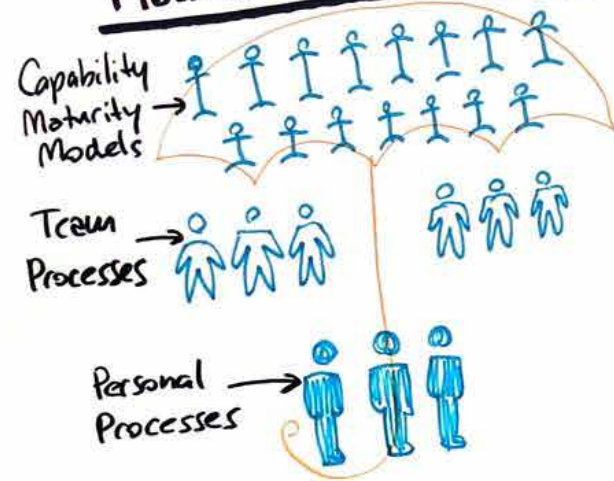


The Rest of the Story: From Product Process to Process Analysis

Jeff Schwalb, NAVAIR

Learn a little,
do a little.

Models & Processes



PROCESS MODELING



The Nouns & Verbs identified become key in the definition of the life cycle models unique to each team.

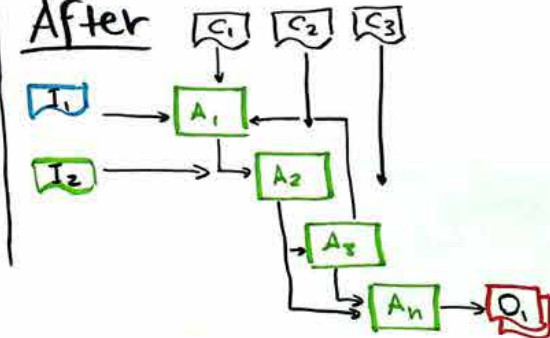
- TPI is based on the application of good engineering in all technical areas, not just software.
- Defining a team's process supports effective planning during launch.
- Effective post mortem analysis relies on well-defined workflows.

Transitioning with Process Modeling

Before



After



ENTRY
TASK
VERIFICATION
EXIT
(ETVX)



Post Mortem Analysis

- Considers artifacts applied during the project
- Looks at data collected throughout project
- Individual Experiences
- Data Minging through automated process tool

Wild, Wild West— How to corral all your developers into creating secure code.

—Jonathan Beck
PNC Financial Services Group



Darwin's Developers

Each team
adapted to their
environment—
different circumstances
different models



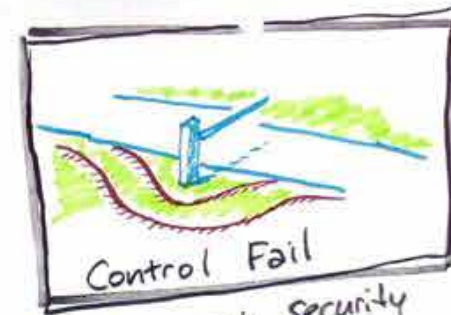
RISK ANALYSIS

- DRIVES SECURITY CONTROLS
- VERY PERSONAL TO AN ORGANIZATION
- RISK FRAMEWORKS
 - FAIR
 - DREAD



THREAT RISK MODELING

- LOOKS AT RISK FROM THE VIEWPOINT OF THE BAD GUY
- NOT JUST SOFTWARE, BUT OPERATIONAL & BUSINESS PRACTICES
- TRUST BOUNDARIES
- CHALLENGING FOR COMPLEX SYSTEMS
- EXPERTISE NEED TO LEAD THE DISCUSSION
- ONLY FOR HIGHEST RISK APPLICATIONS



- Test each security requirement
- Failures need appropriate risk signoff



Best Practice is
the start of your
journey, not the end.

Understand How Each Contribute to making secure software

Original State

- o Static & Dynamic Teams find Vulnerabilities
- o Penetration Test inspect applications
- o Provider generic security requirements
- o Development team fix bugs



Application Security Coaches

- know code, build code
- know all major languages
- focus on remedies
- Sole focused on coaching

- o In setting up training—preferences varied greatly.
- o Instructor-led was deemed most effective

CAMPAIN INCLUDED

- Forum
- awareness articles
- celebrated success YAY!
- Presentations in Senior management
- Executive Support
- o TRAINING PROGRAM
 - customized to each role
 - mandatory & optional
 - CBT & instructor-led

DEFECTS DISCOVERED AT MOST EXPENSIVE STAGE OF DEV LIFECYCLE

APPLICATION SECURITY COACHES

- o understand our developers better
 - o Coding in many languages, in many countries,
 - o Supporting 1000s of app & financial instruments
- NO CENTRALIZATION
NO HEAD**

SECURITY APPLICATION REQUIREMENTS & VALIDATION

- HOW TO PROVIDE MEANINGFUL DIRECTION?
- MOVE TO LANGUAGE & FUNCTIONALITY-DRIVEN REQUIREMENTS
- MAINTAINED AS THREATS ENVIRONMENT CHANGES
- VULNERABILITY SCAN RESULTS FOLDED BACK INTO REQUIREMENTS

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

SGMM

A Zero-depth Entry to Using the TSP:
Julia Mullaney & Summer Fowler



SGMM a management tool for utilities programmatic approach

- ▷ Model
- ▷ Compass Survey
- ▷ Navigation Process
- ▷ Training
- ▷ Partner Program

Quality Planning improved as the product suite advanced



Get People out of the Pool

TSP is not Just for Software

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

- USED LAUNCHES
- REVIEW PROJECT STATUS (Post Mortem)
 - Major Differences in Launch Process
 - Team roles functional
 - Used MS Project for planning
 - Used several cost planning tools

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

LOAD BALANCING

- Made sure tasks matched To committed.

Budget ANALYSIS

Analyzed data from three approaches to finalize the plan.

- Budget Quarterly ^{Course}
- Review Weekly ^{Corrections}
- Reconcile Monthly

DESIGN- LESSONS LEARNED

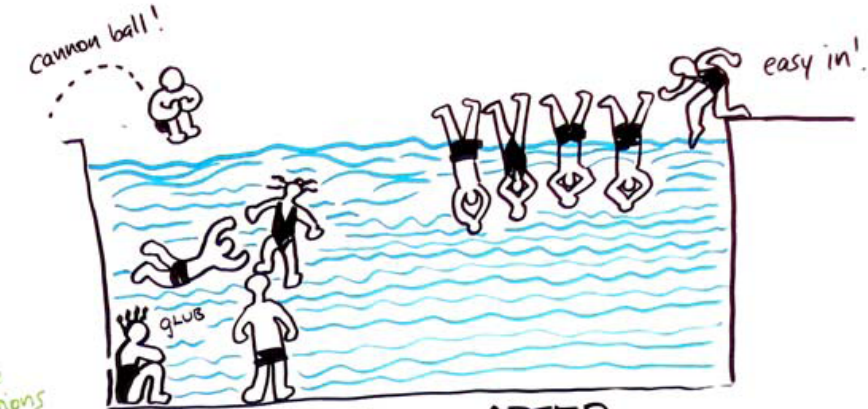
- How to design TSP process to the products produced.
- defined usage, audience

Helped Design Navigator Course, Training Course

TSP set up 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



BEFORE

- o thrown in
- o Sink or Swim
- o every man for himself

AFTER

- o coordination
- o integration
- o enjoyment

And they lived happily ever after

Jesse Schell
CEO, Schell Games

TSP SYMPOSIUM 2014 DAY TWO

INFORMATION FLOW

The secret to Studio Structure

Game Industry 20-30 billion
Lots of badly managed teams

- structure
- how to manage
- miscommunicate with client
- too many meetings
- low morale
- game late
- game SUCKS

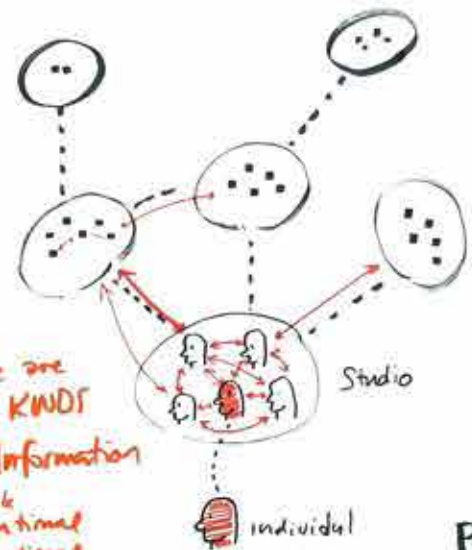
"How do ants & bees make cool stuff?"

Even hives have information flow issues

Cosmologists beginning to think matter & energy reflect cosmic information

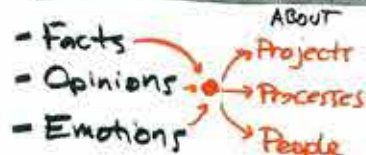
Let's look at what we do through the lens of information flow

You really can smell fear



- There are ALL KINDS of Information
- Work
 - relational
 - emotional
 - history
 - etc

Kinds of Information



How Information Moves



E-mail Conventions

In small groups, "send to all" is not an issue. In large groups, it is.



Office Rituals

A coffee pot becomes a reason to meet.

FREE FOOD BLOCKS INFO - FOOD RUNS FORM MINI-BONDING MOMENTS

SHARING A MEAL CREATES A BOND

WATCH WHO EATS WITH WHO

SPATIAL LAYOUT IS A FORM OF INFORMATION ARCHITECTURE

- Organize by Craft or Project?
- Who gets a door

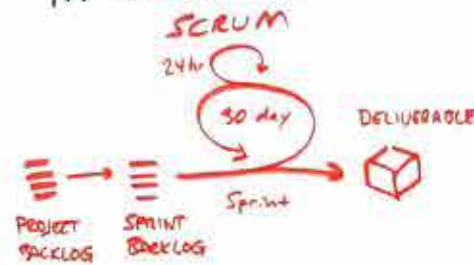
A little distance can cause a big disconnect.

PIXAR DESIGNED THEIR SPACE TO ENCOURAGE PEOPLE TO CROSS PATHS.

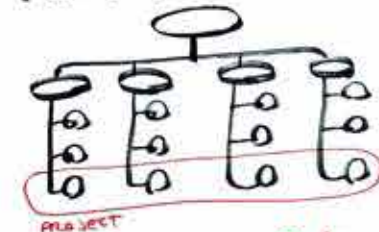
GAME PROGRAMMERS: USE AGILE 30%-ish EXTRA FLUID ENVIRONMENT

MEETINGS

People hate them because they are poorly designed. Standing Meetings automatically timed - your legs tell you when it's time to end.



MATRIX MANAGEMENT



You Report to Project Lead & Division Manager

TWO GODS OR TWO PARENTS?

WHAT BLOCKS INFO FLOW?

- NOISE
- Interruptions
- Lies
- Ruts
- Secrets
- Fear

BEEMANS MEDDLING MATRIX

IT WORKED	I DID ALL 3 GOOD	MEDDLING
I'M USELESS	I SHOULD HAVE MIDDLE	NO MEDDLING

OFFICE POLITICS IS BLOCKED INFO

Casciaro & Lobo's Matrix



People prefer to work with the loveable fool - managers likely to hire the jerk. Organizations that favor the loveable fool more likely to succeed! plays the role of an emotional hub, sustains the emotional network.

COLLECTIVE INTELLIGENCE

- Social Sensitivity
- Conversational Turn-taking
- Number of women in the group

MIXED GROUPS MAKES A BIG DIFFERENCE

THREE CHARTS

ORG CHART	PROJECT	AFFECTIVE HUBS
FACTS & OPINIONS ABOUT PROCESSES PEOPLE	FACTS & OPINIONS ABOUT PROJECTS PEOPLE	EMOTIONAL

KINDS OF JERKS - THEY BLOCK INFO

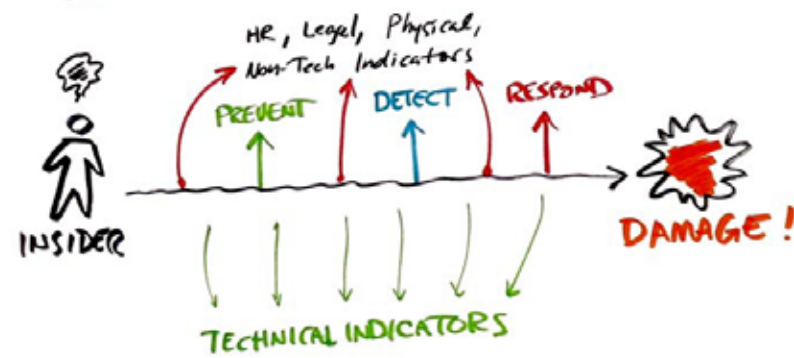
- Inaccessible - will share info
- Unreliable - can't identify info
- Rigid - will recognize info
- Disrespectful - will share info
- Vague - can't identify info
- Unfair - will recognize info

INSIDER THREATS IN THE SOFTWARE DEVELOPMENT LIFECYCLE

DANIEL COSTA & RANDALL TRZECIAK

"Insider used to mean 'trusted', Now connotation is negative"

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



INSIDER THREATS INCLUDE MALICIOUS & UNINTENTIONAL DAMAGE:

MANY EMPLOYEES TAKE SENSITIVE INFORMATION WITH THEM WHEN THEY CHANGE JOBS.

AT A CERTAIN LEVEL, MALICE BECOMES INDISTINGUISHABLE FROM INCOMPETENCE

WHAT'S MOST IMPORTANT TO YOU TO PROTECT?

ACTOR / TARGET / IMPACT
WHO WHAT HOW

"Not everyone is a threat to everything."

PHASES of SOFTWARE LIFECYCLE

REQUIREMENTS DEFINITION

- neglected to define authentication
- neglected to define security requirements
- neglected to define automated data integrity checks

SYSTEM DESIGN

- lack of security in automated workflow processes
- insufficient separation of duties
- insufficient consideration of vulnerabilities by "authorized system overrides"

SYSTEM IMPLEMENTATION

- lack of Code Reviews
- inability to attribute actions

SYSTEM DEPLOYMENT

- lack of enforcement of documentation & backup
- use of same password file
- Unrestricted Access
- Lack of configuration control & well-defined business processes

SYSTEM MAINTAINENCE

- Lack of code reviews
- Ineffective configuration control
- Insufficient backup practices
- End User access
- Ignoring known vulnerabilities



NOT ALL INSIDERS ARE MALICIOUS...

INSIDER THREAT IN SPACE...



UNDER N: Acceptance to Delivery in n hours

Umashankar Velusamy



Not All Deliveries
Are Alike

Think of it as
building an ER
for software delivery



Under-N

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

- Driven by business values
- Doesn't compromise quality
- Co-exists with other business delivery cycles
- Many teams already do it!

- ER**
- Create a framework
 - Use templates
 - Create Governance Process
 - Enable a Medium to handle Requests
 - Institute a support structure

Invoke the Framework
for each Under-N
Capability

FRAMEWORK:

Create a Template with
Key info associated with
the Under-N capability,
both the info needed &
the info to be used.

- Ask the right questions
to the IT teams

GOVERNANCE:

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

CAPABILITY/ INVOCATION:

1. Client invokes a defined
Under-N capability
2. Validate Pre-requisites
3. Accept / Deny
4. POCs follow Template Steps
5. Confirm Delivery in N-hours
6. Measure Business Value

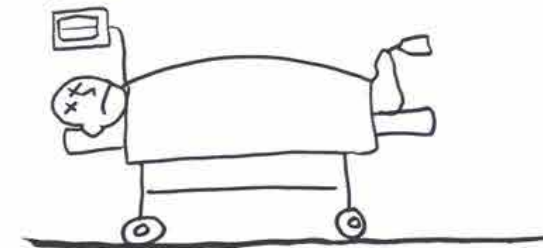
MEDIUM:

- Central Information
Radiator Portal

SUPPORT STRUCTURE:

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

When does it fail?



- o When used to bypass
processes
- o When cutting corners
- o Wrong choice of tools
or Architecture
- o When trying to
squeeze large Projects

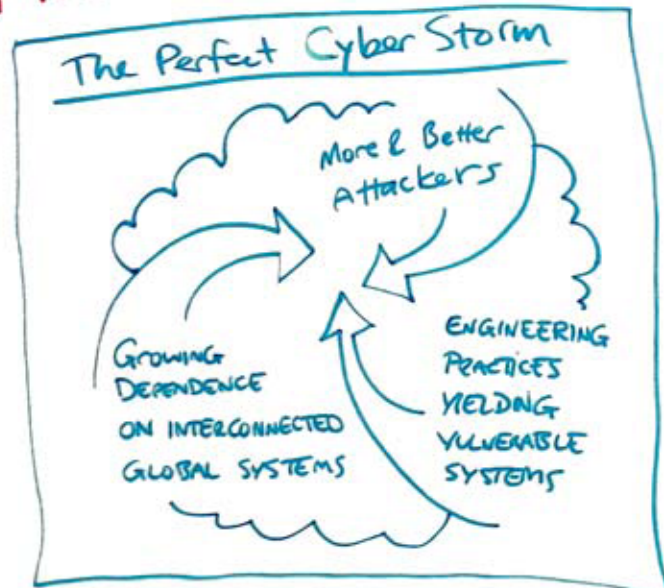
WHEN YOU BUILD IT, THEY WILL COME

Rich Pethia

Our systems are under CONSTANT ATTACK

Credit Card Bad, Identity Theft harder on the individual

CYBER CRIME THE BIGGEST THREAT TO OUR DEFENSE, ECONOMY, & LIBERTY



All aspects of the internet are growing dramatically —
1969: 4 hosts → Today: Over a billion

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

CYBER SECURITY IS ONLY GOING TO BECOME MORE CRITICAL

o SOFTWARE COMPLEXITY INCREASES VULNERABILITY TO ATTACKS

o Open source supply chain is vulnerable

75% of organizations rely on open source as the foundation of their applications —
- Security skills haphazard among developers
- no provenance of code
- no process for updates — transitive vulnerabilities

More than 81% development organizations do not coordinate their security practices across the board

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

A community of Adversaries

- Writers - Make & Sell Malware
- First Stage Abusers - Gather Information
- Middlemen - Collect & Process/Sell blocks of info
- Second Stage Abusers - Do the real damage - ID Thieves, extortion

Organized Crime & Terrorist Networks Involved

The solution is:

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 a software-reliant systems

Design weaknesses drive vulnerabilities
76% of Most Dangerous Errors stem from Design Weaknesses

Static testing & 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.

There's more to it than cryptography

Malware Samples are accumulating at more than a million a month

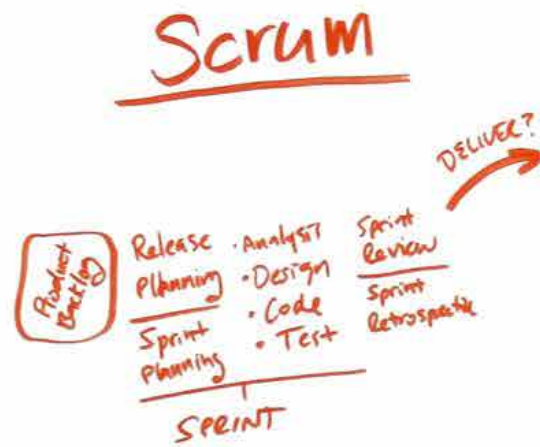
SCRUM:

Creating Great Products and Critical Systems – what to Worry About, What's Missing & How to fix it.
- Neil Potter

Agile ~
a collection of development methodologies
Scrum is ^{one of} the most popular

Agile Manifesto

- Individuals & interactions
 - Working Software
 - Customer Collaboration
 - Responding to Change
- processes & tools
 - comprehensive documentation
 - contract negotiation
 - following a plan
- Agile values the left list more than the right.
- Can we find a balance? BASED ON GOALS AND CHALLENGES



SCRUM HAS BENEFITS

- Work chunked
- Scope changes managed
- Good momentum & early feedback
- easy to learn
- fast feedback
- Opportunities for reduced risk

SCRUM DOESN'T DO EVERYTHING YOU NEED AUTOMATICALLY!

- Plans
- standards
- Requirements

SCRUM SUSCEPTABLE TO AMBIGUITY AND OVERSIGHTS INTRODUCED IN THE REQUIREMENTS & ARCHITECTURE DESIGN

DON'T MISTAKE SPEED FOR PROGRESS

Get good Requirements
Elicitation an art.
- user story

Missing Architecture/Design
Missing Final System Test/Validation

Why Design?

- Identify Possible Problems sooner
- Find Errors earlier
- Clarify Concepts & definitions - be able to communicate them

Better work before the SCRUM gives better feedback during the SCRUM

Plan Ahead: Incorporate Design, Testing, System Testing, etc into the Scrum plan - "Sprint N+1": 60% Design 40% Coding/Test

SCRUMBUT!*

Teams picking & choosing parts of Scrum process.

Does this explain why all the meetings are standing?

* as in, "We do scrum, but we don't do —"

YOU CAN ADD QA WITH GATES & GOVERNANCE

Not all Agile/Scrum teams actually do Agile/Scrum (Ask what they do)

Don't be afraid to add practices - (Just don't break the point of Scrum)



WHEN SCRUM IS DONE WELL, WHAT DO YOU CALL IT?

- SCRUMMY?
- SCRUMPTICIS?