

TSP History and Evolution at Cadence Design Systems

A. Lee Gazlay, Elias Fallon
Cadence Quality Initiative Team
November 2014

Agenda

- About Cadence
- Cadence History with TSP
- Results
- Challenges
- Evolution

Cadence

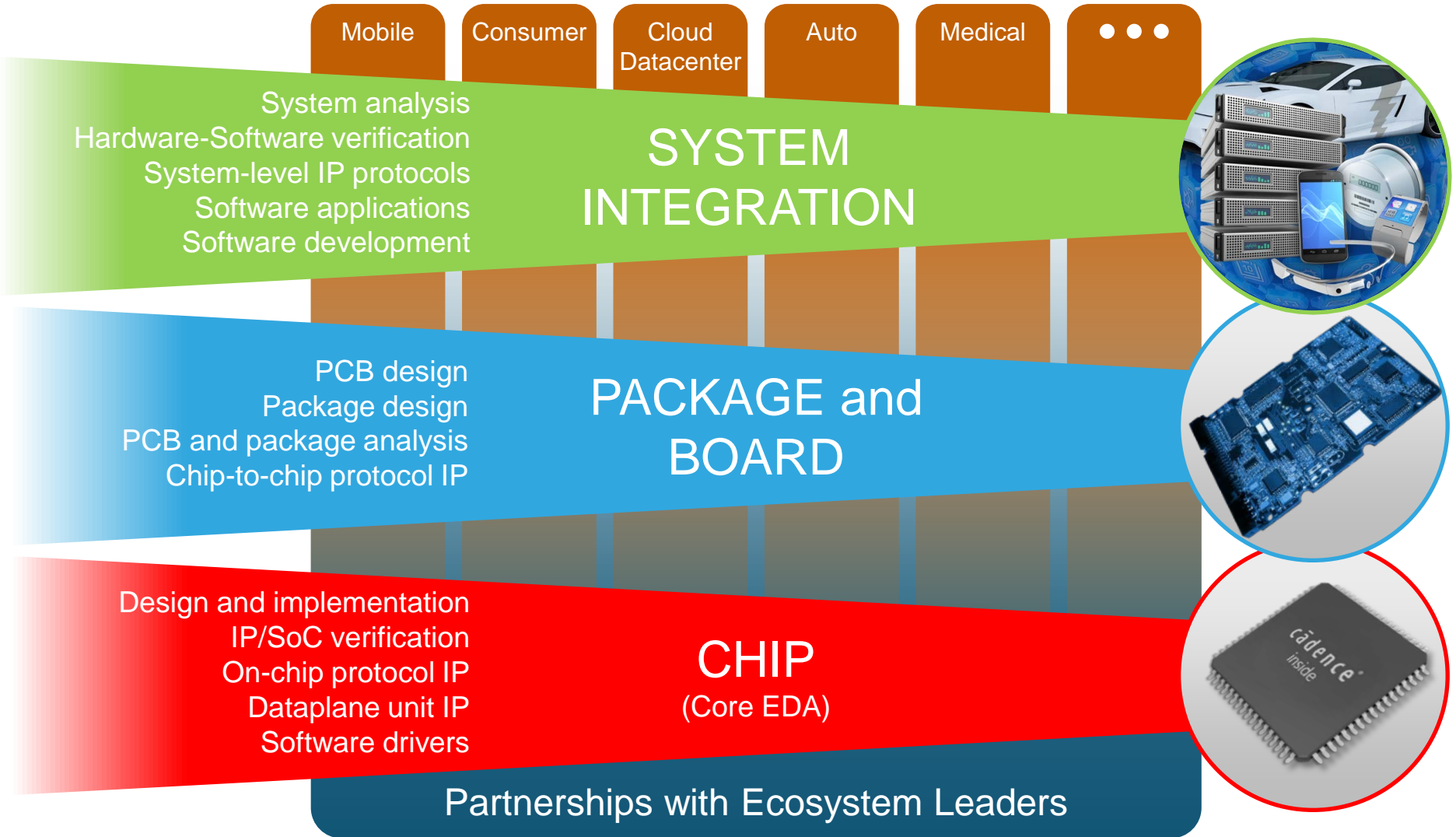
Who we are

▶ A global leader in electronic design automation

- Our software is used to design electronic products like semiconductors, mobile devices, and “cloud” infrastructure
- Dynamic electronics industry creates growth opportunity
 - Grow core EDA business by delivering differentiation and value
 - Grow in new areas, expand portfolio
 - System companies expanding EDA TAM
- Industry leader in technology and innovation
 - Advanced-node, low-power, verification
 - Analog, mixed-signal, 3D-IC
 - IP, emulation, advanced PCB design

Cadence delivers System Design Enablement

From end product down to chip level



Commitment to innovation

3,300

R&D ENGINEERS

1300

FIELD APPLICATION ENGINEERS



R&D INVESTMENT IN 2013

\$534M

PATENTS WORLDWIDE

1500+

Four business groups

LIP-BU TAN
President and
Chief Executive Officer



IPG

IP
Group

Martin Lund

Design IP
Verification IP



SVG

System &
Verification
Group

Charlie Huang

Advanced
Verification
Solutions

Hardware
System Verification

System Level Design



DSG

Digital & Signoff
Group

Anirudh Devgan

Digital Front-End
Design

IC Digital
Implementation

Silicon Signoff and
Verification



CPG

Custom IC &
PCB Group

Tom Beckley

Silicon, Package,
Board

Custom IC
Implementation

Custom Simulation

Why Did We Start Using TSP?



Software Debt:
It's like digging a hole.

- 50% of our R&D resources are dedicated to servicing our S/W debt

- The Software industry is the only modern high-tech industry that relies primarily on testing to improve quality
- Most software defects are found in or after test when defect removal costs are the highest and least effective

Economics of Software Quality

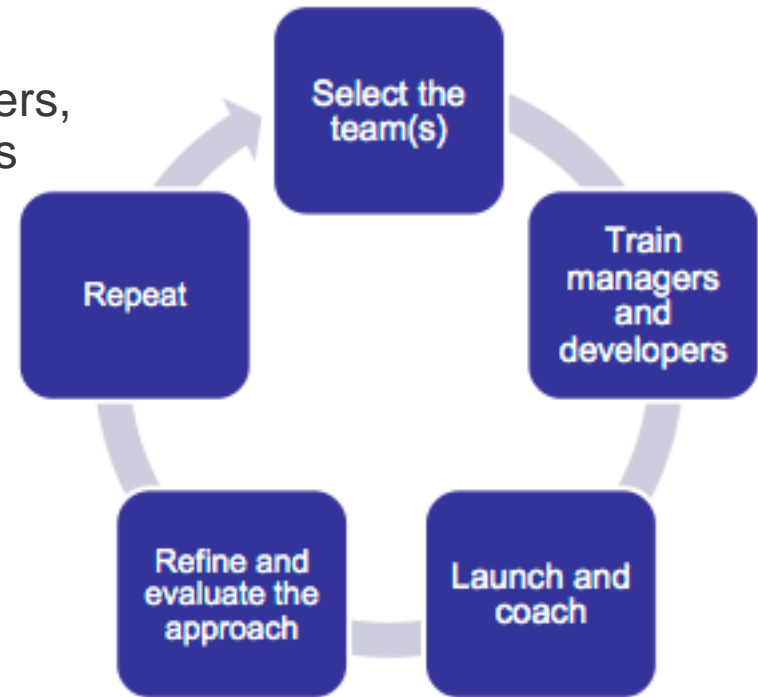
	Efficiency	Effectiveness	Predictability
	Avg. removal rate (defects/hr)	Phase yields (% of defects removed)	Estimated effort
Design Review	1.5	50% to 70%	Low variability - based on product size
Design Inspection	0.5	50% to 70%	
Code Review	4	50% to 70%	
Code Inspection	1	50% to 70%	
Unit Test	0.2	35% to 50%	High variability - based on time to find & fix defects
Integration Test	0.1	35% to 50%	
System Test	0.05	35% to 50%	

Software Engineering Institute | Carnegie Mellon
Team Software Process
© 2011 Carnegie Mellon University

- Testing quality into our products is not enough
- Shift to earlier defect detection and prevention
- Change requires a framework for change
- TSP is that framework

TSP Implementation Strategy

- TSP is implemented project-by-project
 - Select two pilot teams
 - Train top-down, starting with senior managers, then project managers, then team members
 - Conduct a TSP Launch to kick-off each project
 - Evaluate and fine tune the approach
 - Repeat this cycle increasing the scope at a sustainable pace
- Build the TSP coaching team
 - Part-time internal coaches
- Build internal TSP expertise (instructors, mentor coaches)
 - Use SEI to bootstrap and for additional coaching or instruction bandwidth



History of TSP at Cadence

- Started with 2 pilot teams in 2011

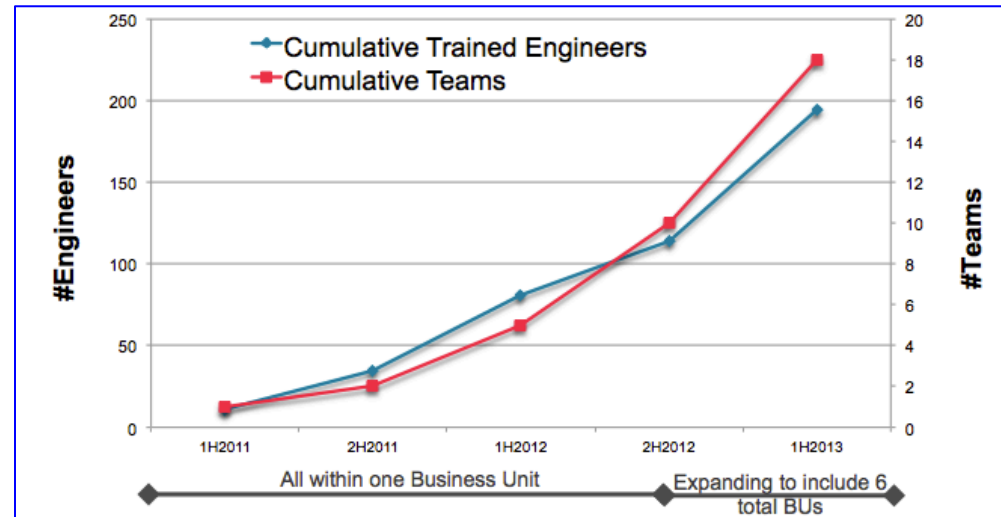
- One team in Pgh, one team in SJ
- Both in the CPG Business Group

- Extended to 8 additional teams in 2012

- All in CPG Business Group
- Added teams in Sophia (France) and Noida (India)
- First distributed teams
- Team sizes: 6-24 engineers

By end of 1H2013:

- 18 teams practicing TSP
- Almost 200 engineers trained in PSP Fundamentals
- Over 30 engineers trained in PSP Advanced
- 2 certified PSP instructors
- 3 certified TSP coaches
- 10 coach candidates in pipeline



- 2013: Cadence Quality Initiative (CQI) created

- Extended beyond CPG BG in 2013

- 6 additional teams in other BGs, and two additional sites

Vision Statement:

Be recognized as the industry leader in delivery of high-quality, full-featured products by 2017

Strategy:

To affect a change in company culture to include a self-sustaining and self-funding quality focus

Tactics:

“Do the right things” – by insisting that effort is prioritized based on overall business value and realistic estimates

“Do things right” – by driving process improvements, tuned for each team’s current state and greatest value

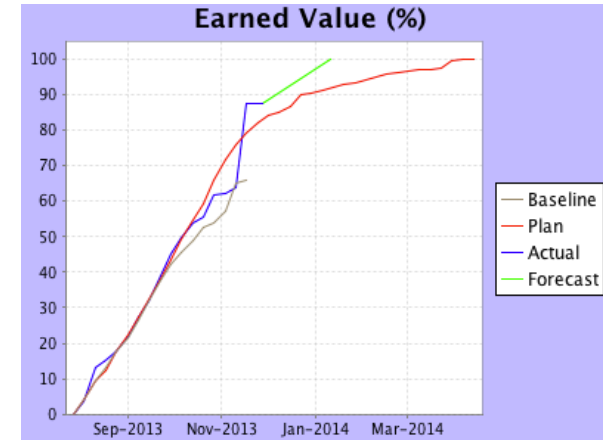
“Do it consistently” – by dedicating bandwidth even when it is difficult

PSP/TSP (one of many things we are doing)

What We Learned

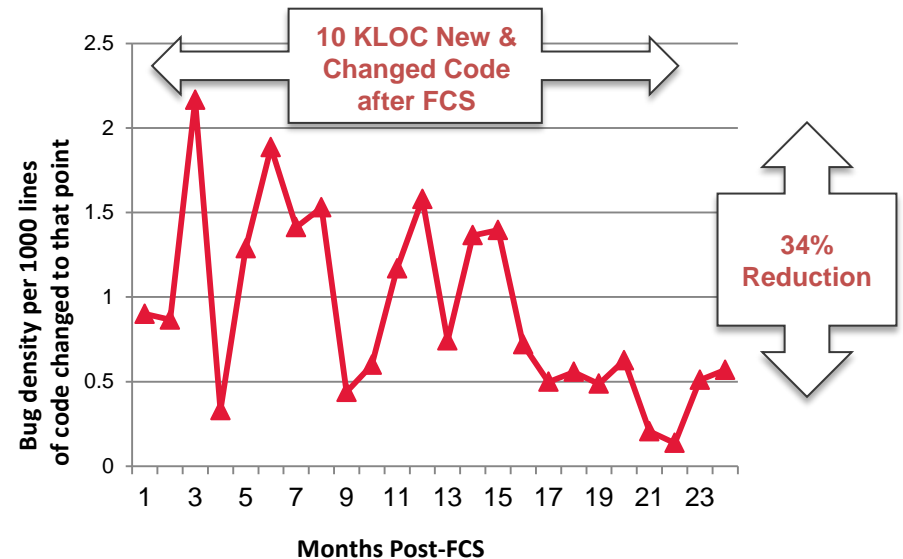
TSP Works!

- Teams are more methodical about
 - Turning requirements into realistic plans
 - Planning for interruptions (like unplanned CCRs)
 - Monitoring where their time is going
- Schedule predictability is typically the first area where teams improve
- Documented processes build awareness within the team and provide the framework for change
- Objectively connected TSP to significant defect reductions
 - 34% reduction in defects / kloc between incremental releases
 - 5x reduction in customer-found defects release-to-release
- Customers noticed the improvement



TSP teams deliver higher quality full-featured products

- It is possible to deliver significant new functionality in incremental releases, and end up with fewer defects than normal
- It is possible to do new development in a base release and have 5X fewer defects than previous releases
- *What would it mean to your team to be able to free 30% of every week for new development instead of fixing defects?*



Release	Project Added & Modified LOC	Pre-Release Defects per KLOC	Defects Within 2 Months of Release per KLOC
R5	5702	25.95	8.94
R6	24770	5.25 <i>5x improvement</i>	1.77 <i>5x improvement</i>

What We Learned continued

There are Issues

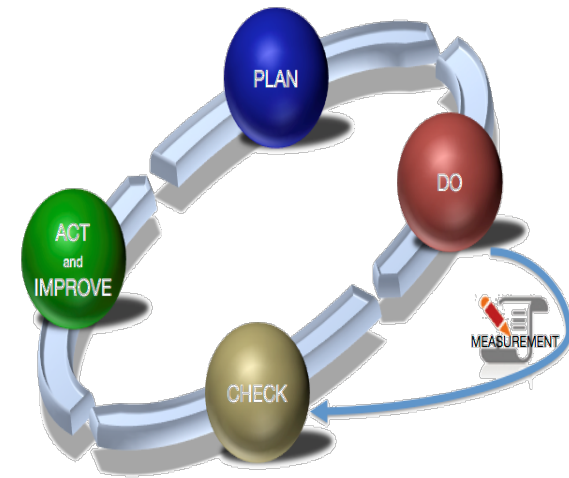
- Our TSP deployment model is PUSH
 - Latest teams are doing TSP because they were volunteered
 - Lack of commitment, engagement and support from middle management
 - Number of engineers who completed training is a poor measure of success
- PSP training has a high cost of entry for many teams
 - 4.5-day PSP Fundamentals class is a prerequisite for the first TSP launch
 - Live, in-class training works best for co-located teams; many Cadence development teams are distributed
 - Experimented with remote and recorded delivery. Worked for PSP Advanced. Did not work for PSP Fundamentals.
 - Engineers learned techniques they were not ready to apply immediately
- Coaching bandwidth
 - Teams with the most coaching attention were the most successful
 - Not ramping up new part-time coaches fast enough

We are not on a trajectory toward success

The Cadence Evolution of TSP

We Used TSP on TSP

- CaDET:
 - Cadence Development Engineering Training
- Developed with the SEI
- Combines PSP and TSP concepts with best practices from other managed development frameworks
 - And addresses our specific issues
 - High overhead of initial training
 - Live instruction for distributed teams
 - Learning skills before you're ready to apply them
 - Coaching bandwidth
- Provides a framework to include other Cadence best practices and tools

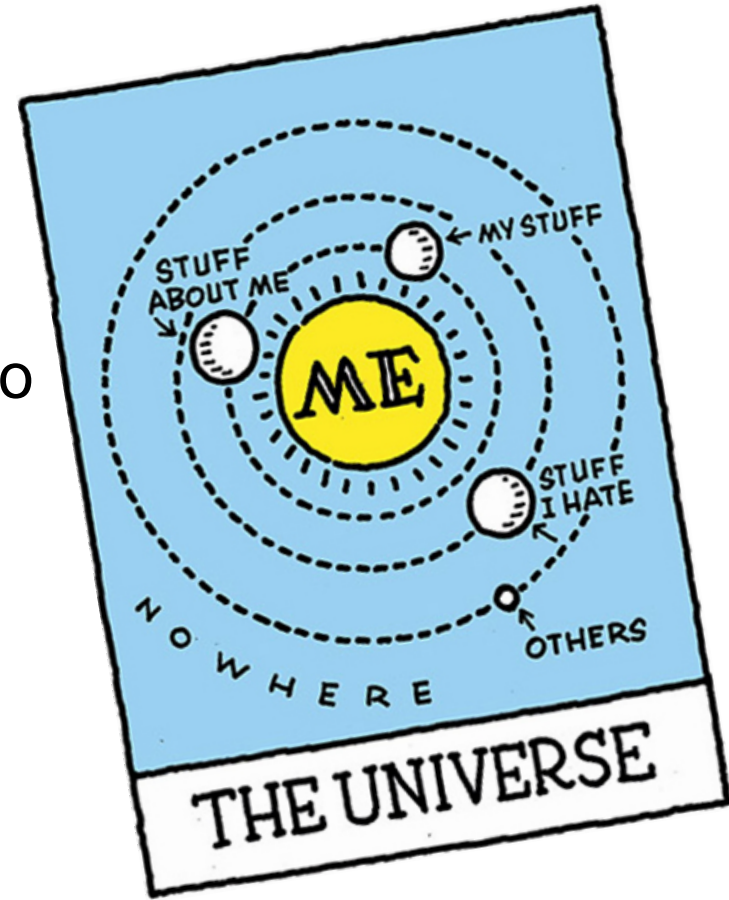


Comparison of TSP and CaDET

	TSP	CaDET
Training	<ul style="list-style-type: none">• 2 Levels of training – PSP Fundamentals and PSP Advanced• PSP Fundamentals requires 4.5 days of live instructor-led training• Students learn material they may not use immediately	<ul style="list-style-type: none">• Multiple courses; 20-minute modules• All computer-based, online training• Five hours of training prior to first launch• Team members take only the applicable modules, when they need them• Modules added per team improvement plan
Launch / Relaunch	<ul style="list-style-type: none">• Team leader and team members have little understanding of what a TSP launch is before experiencing the first one• The TSP coach drives the launch meeting• First launch requires 2-4 days• Re-launches require 2-4 days each	<ul style="list-style-type: none">• CaDET training starts with the launch process; overview for team members, in-depth for team leaders• <u>Team leader is trained to drive the launch</u>• First launch completed in 2 days or less• Re-launches are 1 day or less, including retrospective of previous cycle
Devel Cycles	<ul style="list-style-type: none">• Cycles are 8-12 weeks	<ul style="list-style-type: none">• Initial cycles are shorter, 2-4 weeks• More, brief initial cycles means more opportunities for incremental improvement• Cycle duration can lengthen as teams gain experience

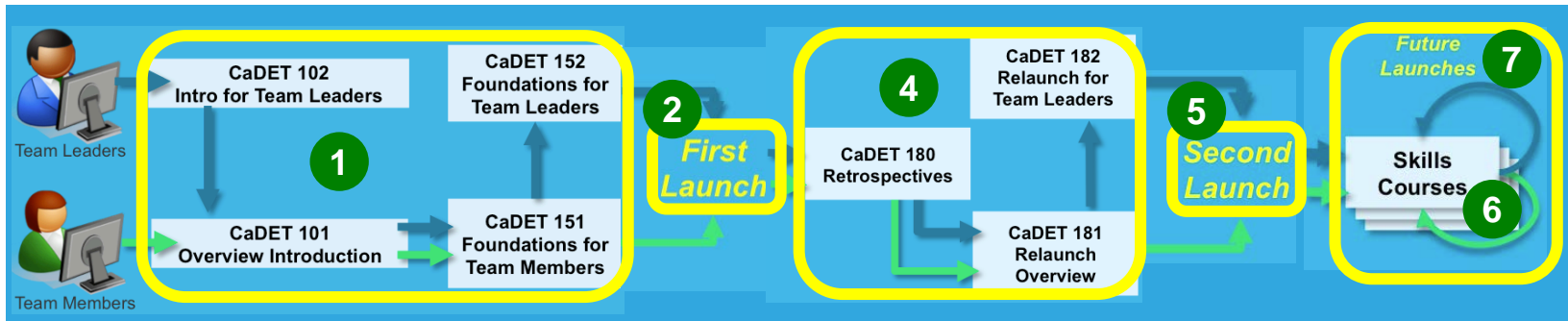
What's In It For Me? [WIIFM]

- Employing a *Managed Development Framework, like CaDET*, allows you to improve any area of predictability
 - Improving estimating accuracy
 - Improving overall quality
 - Better understanding and control over your time, plans and commitment
 - Opportunities to learn and get better
 - Improved teamwork and communication

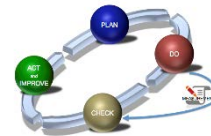


“Having spent more than a decade working in EDA software without any formal process or useful quality metrics, I can say the change to CaDET is a breath of fresh air. Although getting in the habit of tracking time and defects was difficult to begin with, and the idea of someone looking at every line of code was daunting. I find that the quality of my designs and code have dramatically increased, and that I feel more confident in my abilities as a developer.”
- CaDET Developer

CaDET: Learn and Evolve Incrementally



- 1 First, the team completes the CaDET Introduction and Foundations courses
- 2 Next, the team has a basic launch to setup initial processes and plan the work for the next few weeks
- 3 The team applies CaDET skills to their work and collects data
- 4 The team completes the CaDET Retrospectives and Relaunch courses
- 5 The team has a relaunch and uses the data collected in the preceding cycle to plan another cycle
- 6 The team completes additional CaDET Skills courses
- 7 The team applies their CaDET skills and knowledge for additional cycles then repeats steps 6 and 7



CaDET: Measuring Success

CaDET went live Jan 2014

- 695 engineers have completed CaDET-101
- 600 engineers have completed CaDET-151
- 259 “managers” have completed CaDET-152
- 49 teams have completed at least one CaDET launch
- 33 teams (262 engineers) are “**CaDET deployed**”

- Focus on Launch Reports
 - All CaDET teams submit launch reports to a central repository
 - Reports are evaluated against a checklist of required items
 - Incomplete reports must be corrected
 - Teams achieve maturity levels (0-5)
 - Routine reports generated for management
- Routine Management Interviews
 - Standard questionnaire so all interviewers are asking the same questions, getting and giving the same information
 - Allows us to identify areas of misalignment
 - Sometimes teams give feedback to their management they don't give to us
- Central Quality Reporting Tool
 - Standard customer quality and responsiveness metrics

Summary

- TSP Works!
 - This is the *Scientific Method* applied to software development
 - Something most software engineers are never taught (or asked) to do
 - The only way it doesn't work is if you don't do it
- This is not a silver-bullet solution
 - This is fundamental change management
 - There will be natural resistance to this change at many levels
 - Management engagement at every level is essential
 - Summarizing changes in terms of WIIFM is very important
 - It takes time and persistence and alignment
 - This message must be delivered from the beginning and repeated continuously
- You need to figure out what works for you
 - Apply the same principles that TSP teaches:
Plan – Do – Check – Act / Improve
 - We are still iterating: trying new things, evaluating, improving
 - We are not assuming we've made all the right decisions (yet)

cā dence[®]