Honeywell

Introduction to PSP & TSP

Steve Janiszewski
Director Six Sigma Software
Honeywell International
March 21, 2001

Systems & Software Six Sigma Charter

- **Improve the quality and reduce cost and cycle time of systems and software development, freeing up resources and enabling growth**
  - **Monitor, identify, pilot emerging systems & software process technologies**
  - **Deploy proven processes and tools throughout Honeywell**
  - **Drive alignment of Six Sigma, PSP/TSP, and CMM processes**
  - **Provide enhanced Green Belt and Black Belt learning program**
  - **Provide CMM based process assessment to Honeywell sites**
- **Offer Software 6σ services to external customers and suppliers**
  - **Establish external revenue stream to offset cost of Honeywell SPI**
- **Establish Honeywell as the leader in high quality systems engineering & software development**

…Create value and "accelerate the growth imperative" in the Honeywell business environment

Software Process Improvement Vision

- SPI is driven top down by business goals
- Each site can objectively measure SPI's contribution to business goals
- There is a published SPI plan at each site
  - Baselines cost structure and quality levels
  - Identifies targeted process improvements
  - Calculates ROI
- Each site tracks SPI deployment and ROI and manages to target
- Measurable benefits are achieved and provide the basis for a sustainable continuous improvement culture
- Sites achieve quantifiable annual productivity improvements in the 5% - 15% range and set their own SEI level goals each year moving through a progression of levels at an appropriate pace culminating in a sustainable six sigma process at SEI level 5

Six Sigma SW Products & Services

- **Enablers**
  - **Executive and management seminars, program management training tie-ins**
  - **One-On-One management meetings**
  - **SPI workshops, ROI model, mentoring, CMM assessments**
  - **SW scorecard tracking and reporting**
  - **Red program reviews**
- **Technologies**
  - **Requirements Management (training)**
  - **Appraisals & Defect Prevention (training, automation)**
  - **Design (training)**
  - **Software Project Management (training)**
  - **PSP, TSP, 6σ for SW (training, launches, coaching & automation)**

Critical Software Business Needs

- **Software-dependent businesses have three critical needs**
  - **Better cost and schedule management**
  - **Better quality management**
    - **When poor quality software is allowed into test, finding and fixing defects is**
      - nearly half of development costs
      - uncontrolled
      - largely unpredictable
  - **Cycle time improvement**

- **All businesses are becoming software businesses**
  - **Software costs and schedules dominate many business plans**
  - **Software quality limits our ability to field many critical systems**

- **In order to meet these needs, one cannot simply try harder**

One definition of insanity: doing the same thing over and over and expecting a different result

Something different - A control system viewpoint

- The outputs of a system, y are usually a function, f, of a set of control variables, x.

$$y = f(x) + \varepsilon$$

- The y's cannot be controlled directly, only by modifying the x's. Statistical measurements are necessary to avoid re-acting to the noise ε

- For a software project, y is typically cost and schedule and x is product quality and hours on task.
  - Cost and schedule cannot be directly controlled.
  - It is possible to indirectly manage cost and schedule to overall project goals by continuously managing product quality and time on task to appropriate intermediate goals

- Ideally we would like software process that acts like a responsive, "closed loop" control system driving project performance to target

PSP enlists each engineer in pro-active product quality management

SEI & the Capability Maturity Model

- **The SEI Process Program was created in 1986 to improve the practice of software engineering by improving the software engineering process.**

- **History**
  - **Process Maturity Framework - 1987**
  - **Software Process Assessment - 1987**
  - **DoD Software Capability Evaluation - 1987**
  - **SEI Capability Maturity Model - 1991**
  - **Personal Software Process - 1995**
  - **Team Software Process – 1996**

CMM, PSP, & TSP

Capability Maturity Model (CMM)

- The CMM is a conceptual framework based on state-of-the-art software engineering practices that help software organizations to
  - characterize the maturity of their processes
  - establish goals for process improvement
  - set priorities for immediate action
  - envision a culture of software engineering excellence

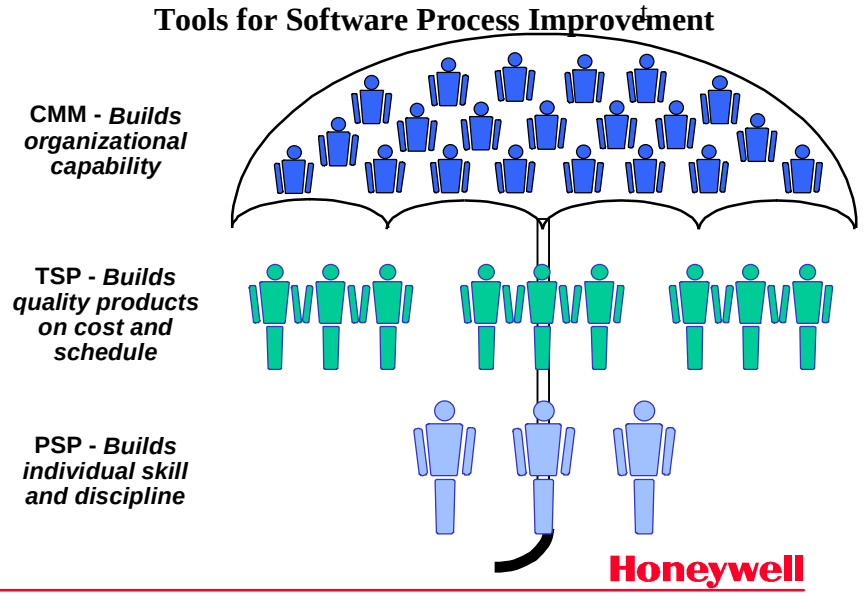
Team Software Process (TSP)

- The TSP adds a project management layer to the PSP
- It address performing software development and maintenance using high performance inter-disciplinary work teams
- It is a level 5 process for managing project teams of 5-10 engineers
- It can be extended to larger projects using TSP multi-team

The Personal Software Process (PSP)

- The PSP is a level 5 process designed for cost effective individual use.
- It applies to most structured personal tasks.
  - developing program modules
  - defining requirements or processes
  - conducting reviews or tests
  - writing documentation, etc.
- PSP augmented TSP can support the development of large-scale software systems
- It can be used to accelerate an organization from level 2 to level 5

PSP, TSP, & CMM

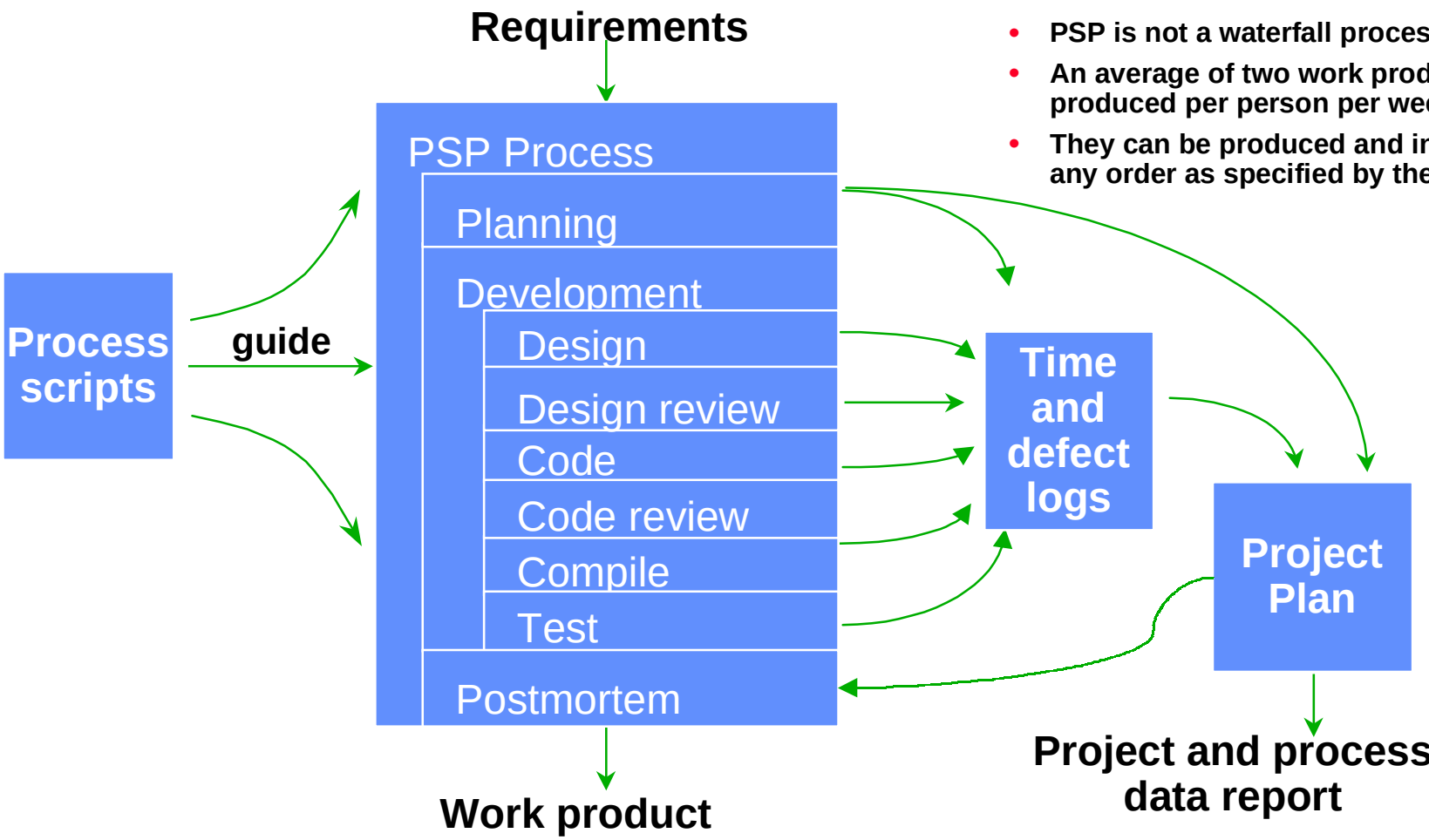


PSP, TSP, and CMM Compliance

Level	Focus	Key Process Areas (KPA)
5 Optimizing	**Continuous process improvement**	√ **Defect prevention** √ **Technology change management** √ **Process change management**
4 Managed	**Product and process quality**	√ **Quantitative process management** √ **Software quality management**
3 Defined	**Engineering process**	√ **Organization process focus** √ **Organization process definition** Training program √ **Integrated software management** √ **Software product engineering** ✯ **Intergroup coordination** √ **Peer reviews**
2 Repeatable	**Project management**	Requirements management √ **Software project planning** √ **Software project tracking** Software quality assurance Software configuration management Software subcontract management

√ *indicates the CMM KPAs that are addressed at the personal level by PSP and at the team level by TSP*
★*indicates the CMM KPAs that are addressed by TSP*

The PSP Process Flow



- PSP is not a waterfall process
- An average of two work products are produced per person per week.
- They can be produced and integrated in any order as specified by the TSP plan.

- PSP training address implementation only
- The PSP process is applicable to analysis, architectural design, integration & test, documentation production, etc. by substituting different development activities, changing size metric, modifying estimating algorithm

PSP Metrics

- There only three basic metrics in PSP.
  - effort, measured in minutes
  - defects found in the product
    - Fix time, type, injection phase, removal phase, description
  - product size, measured in lines of code (LOC)

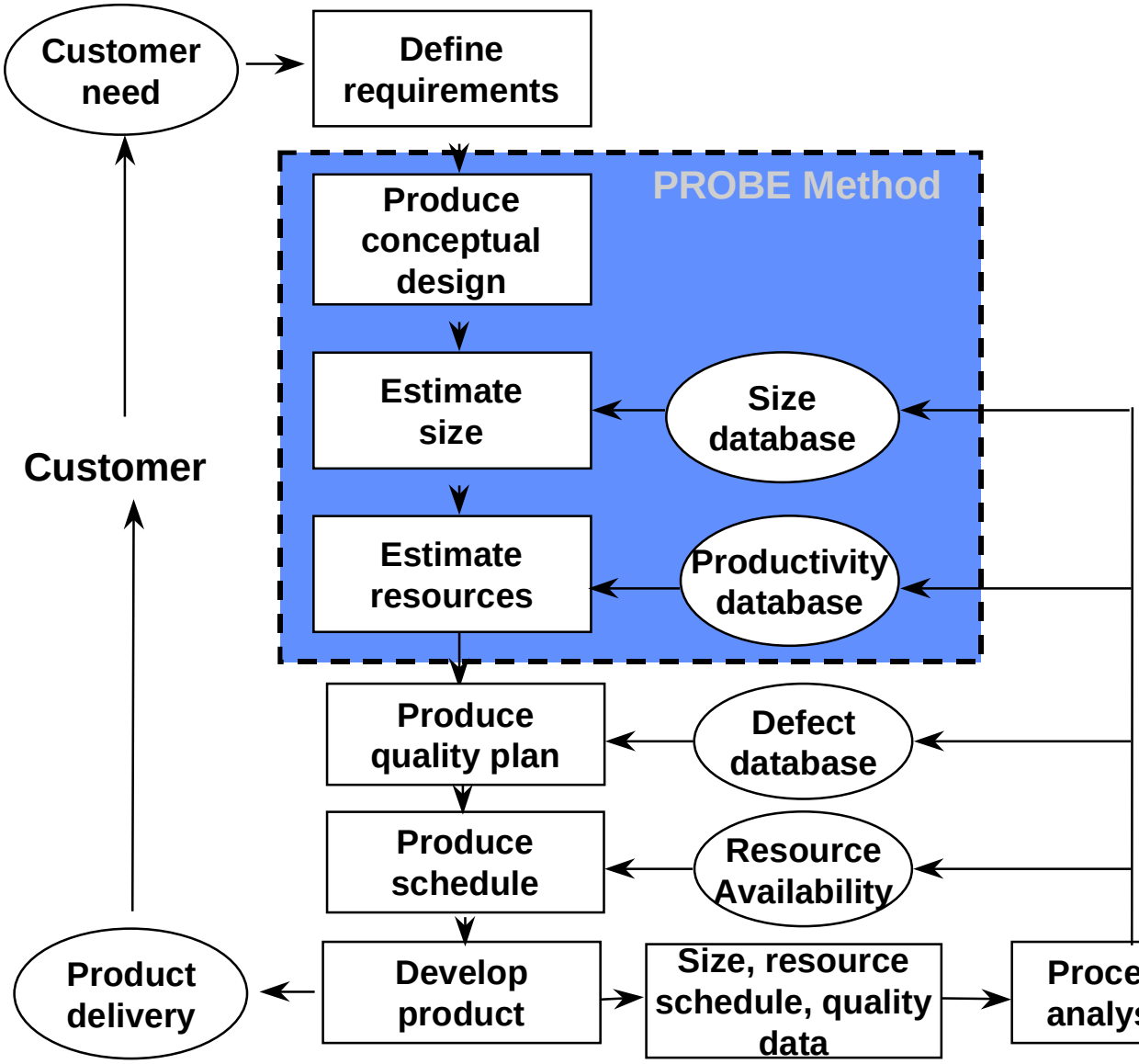
- At Honeywell, data on these measures are recorded in-process using an automated database

- Data recording overhead is exceptionally low – typically less than 5 minutes per day

- Engineers are provided with real-time data analysis for decision support in the course of doing their task and performing a task post-mortem

Data must be regularly used by the person collecting it. Otherwise data collection will stop!

PSP: A Closed Loop Process



- Individuals plan their tasks based on their own historical time, size, and defect data. This provides more accurate estimates since individual performance is the single greatest source of measurement variation
- Individuals log time and defect data in process as they perform their tasks
- Individuals manage their own tasks using real-time feedback provided by the difference between planned and actual process metrics
- Activities are driven to planned performance
- Planned performance levels serve as phase exit criteria
- Automated in-process data acquisition and real time analysis is a key enabler

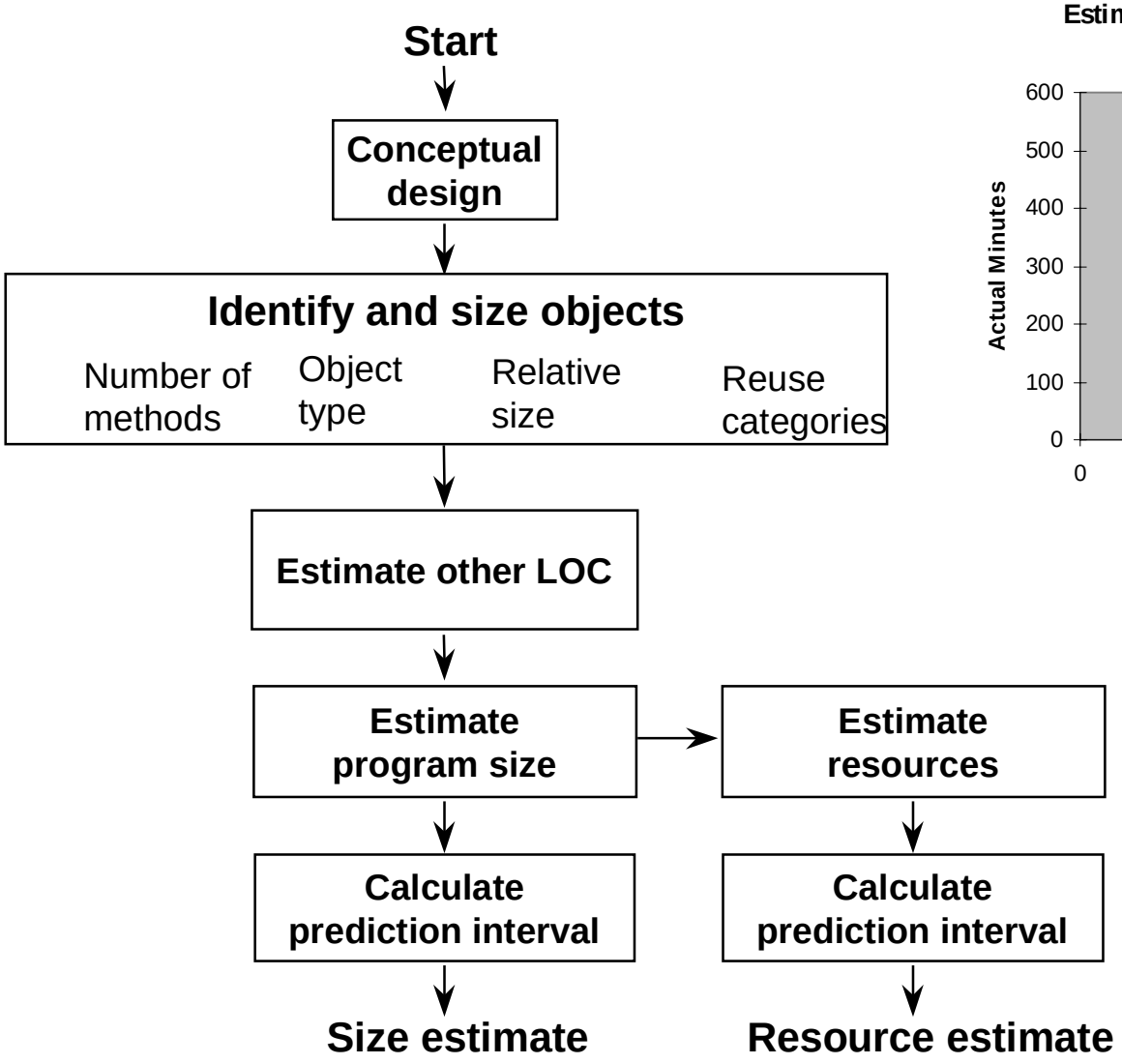
The PSP Estimating Strategy

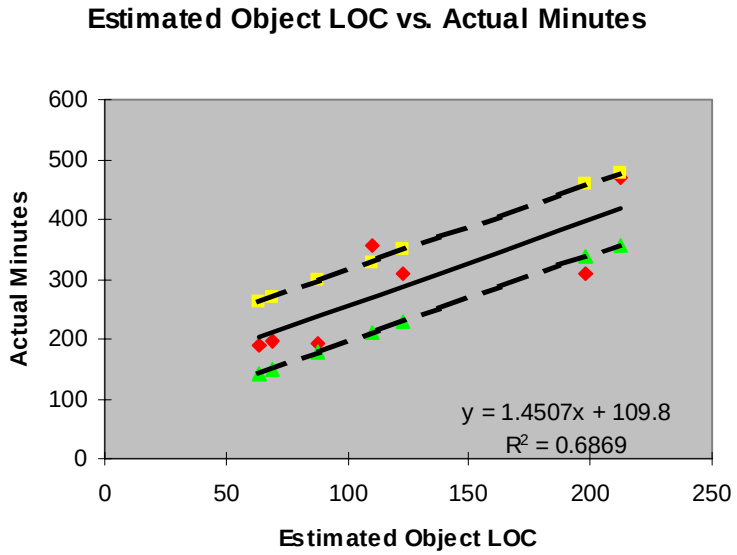
- Individual engineers
  - estimate their tasks to a granularity of several days
  - base these estimates on their own data
  - calculate prediction intervals for their individual estimates
  - combine individual estimates into project estimates and prediction intervals into a project level prediction interval
- This results in
  - more accurate estimates
    - estimating algorithms require less historical data and are more accurate when calibrated to an individual rather than a group - individual data frequently correlates when group data doesn't
    - combining estimates from multiple sources tends to average out estimating bias
    - 1σ prediction intervals RSS when they are combined yielding a $\sqrt{n}$ improvement relative to a single estimate
  - engineers who are more committed to the estimates

Task Granularity

- PSP works with highly granular tasks – typically each engineer completes an average of two per week
- During planning, high task granularity yields better estimates
  - For a 1000-hour job,
    - if estimating accuracy is + or - 50%
    - the estimate range is from 500 to 1500 hours
  - In 25 parts, each with 50% error,
    - the total would be 1000 hours, as before
    - the estimate range is between 900 and 1100 hours
- During tracking, high granularity allows better trend analysis and dynamic load balancing
- Finally, high granularity yields good statistical data in a short time and provides rapid quantitative feedback for process improvement

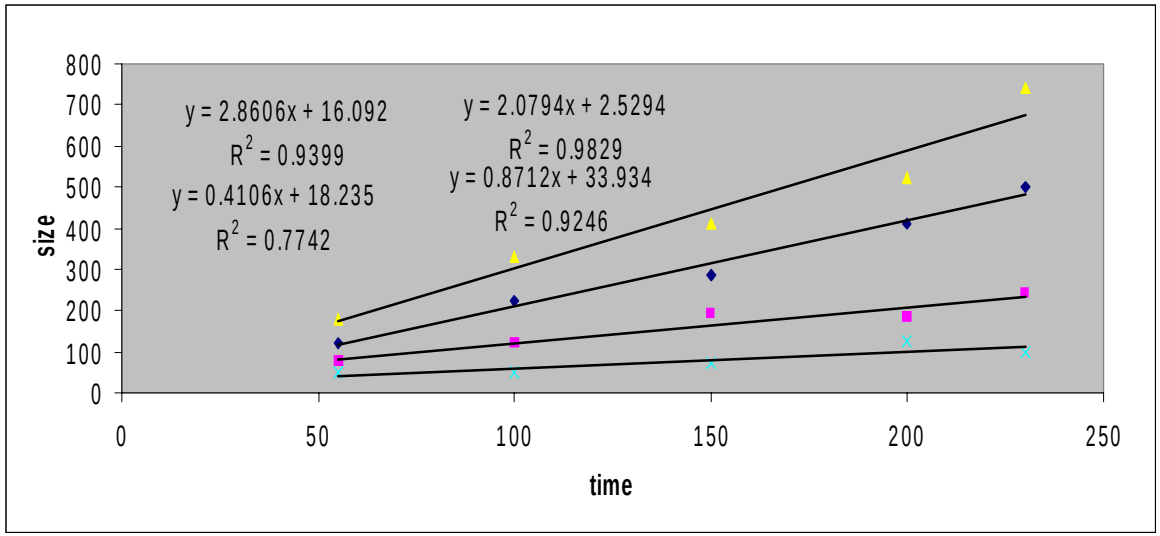
The PROBE Estimating Method

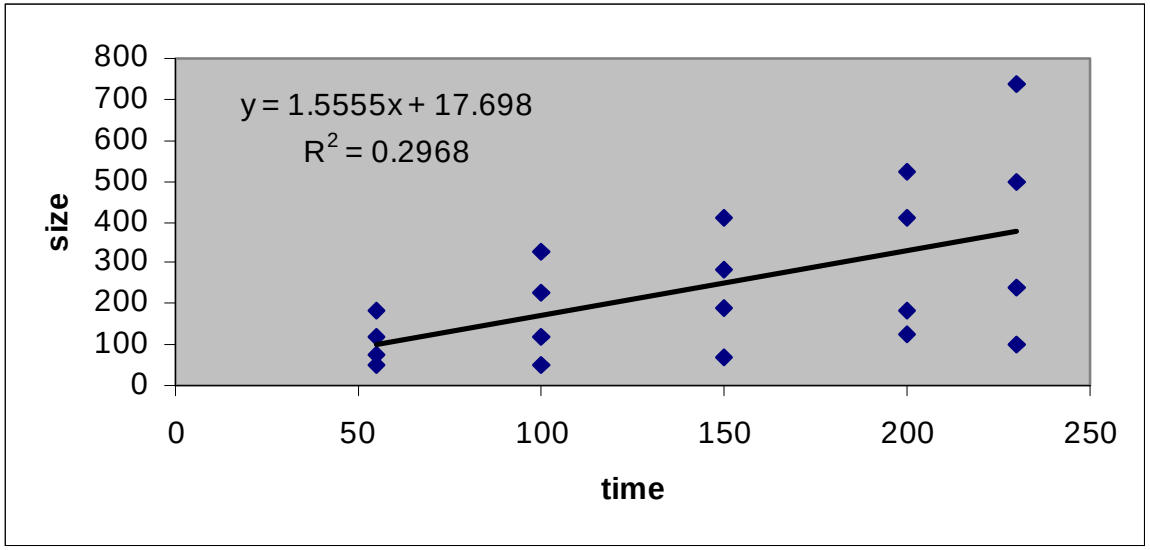




- To estimate resources, PROBE uses the historical relationship between estimated object LOC and actual resources
- The prediction interval (PI) is the range around the estimate within which the actual result is likely to fall.
- The PI is computed using a 70% likelihood (probability).

Grouping Data Can Destroy Correlation





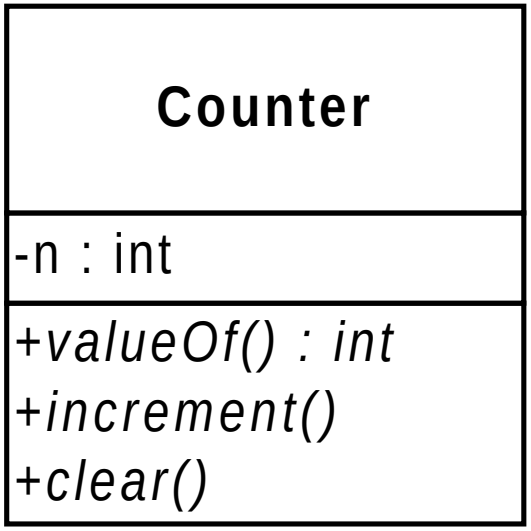
- **Estimating data for 4 individuals**
  - **each has excellent size time correlation**
  - **each is predictive**
- **Aggregated data**
  - **correlation has been destroyed by grouping**

PSP Design

Object Specification	Internal	External
Static	Logic specification	Functional specification
Dynamic	State specification	Operational scenario

- PSP includes a separate design phase that produces its own work products and imposes standards for design content
- Distinct design work products
  - foster identification of exception processing prior to writing code
  - Help eliminate redundant code
  - allow systematic orthogonality and completeness checks
  - can be inspected for high risk defects prior to writing code
- Design is viewed as a defect prevention activity

Functional Specification



valueOf
Return n
Increment
n = n + 1
Clear
:: n = 0

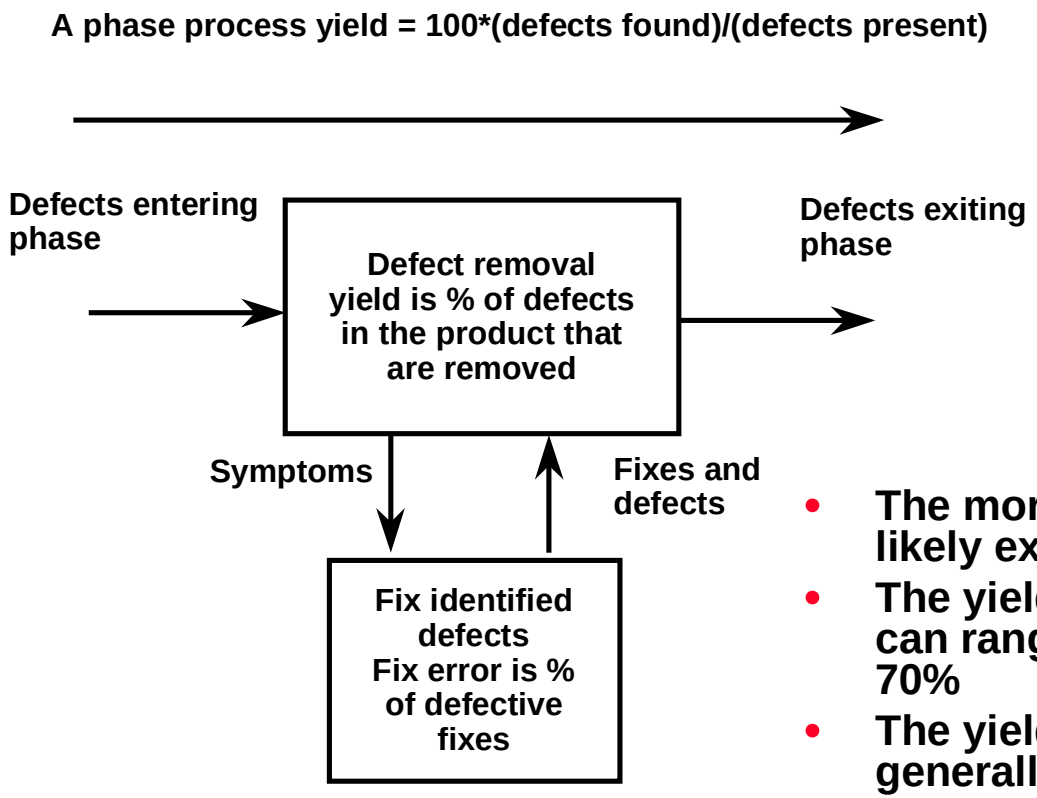
Now?

valueOf
:: Return n
Increment
n = max :: Return ||
n < max :: n = n + 1
Clear
:: n = 0

How about now?

Adequate to code?

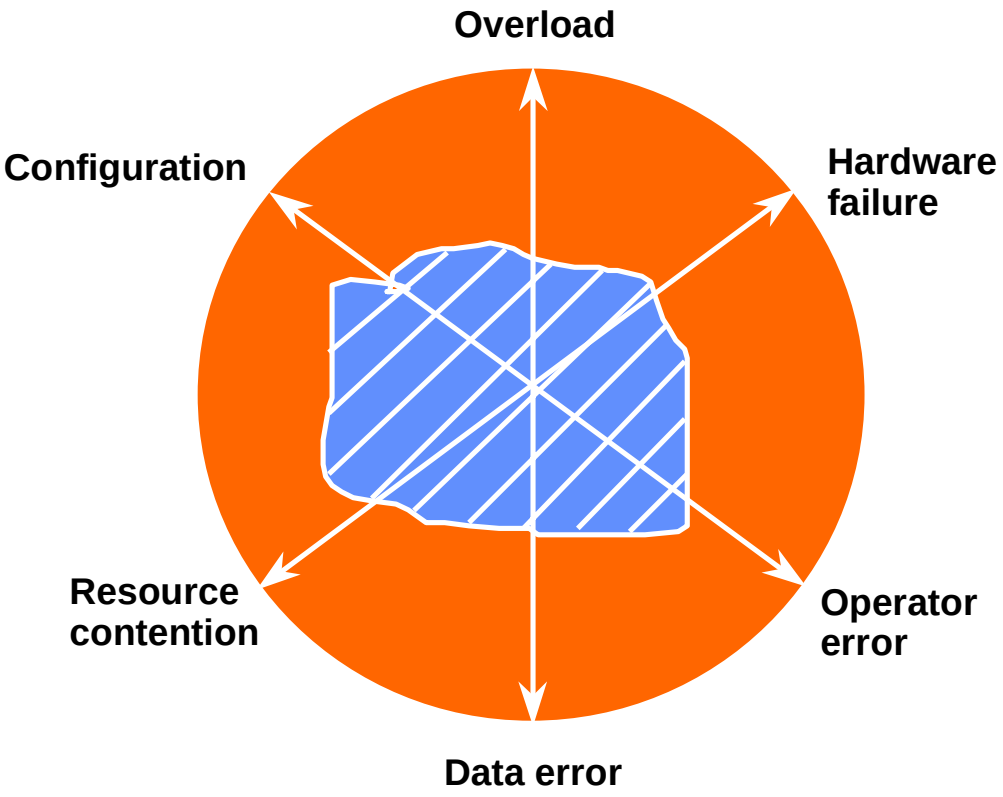
Defect Removal Activities as Filters



- **Defect removal methods**
  - **Appraisals**
    - **Walkthroughs**
    - **Inspections**
    - **Reviews**
  - **Compilation**
  - **Testing**

- **The more defects enter a phase, more will likely exit.**
- **The yield of an individual appraisal activity can range from 50% - 80% with an average of 70%**
- **The yield of an individual test activity is generally less than 50%.**
- **The later a defect is removed, the higher its removal costs.**

Testing



Safe region - tested (shaded)

Unsafe region - untested (unshaded)

- Finding defects with testing takes time and is expensive.
- Even many simple programs cannot be exhaustively tested. The input domain is only sampled over a relatively small area.
- The number defects removed in test from the sampled area is a predictor for defect density associated with the remainder of the input domain.
- Testing is like clearing a path through a mine field: travelers are only safe on the cleared pathways.
- Disciplined engineers use reviews and inspections to clear the entire mine field.
- Test data provides real time feedback on review effectiveness and a source of personalized checklist data
- High defect density in test is an indicator that there are process problems

Inspections

- Peer inspections are not part of PSP but are normally include in TSP
- Personal reviews are done prior to team inspections
  - less “noise” to distract reviewers
  - reviewers can’t fill their quota with easy defects
  - reviewers focus on requirements and interface defects
  - inspection teams tend to be small and are composed of immediate customers for the product
  - checklist should be orthogonal to personal reviews
- Inspection teams are kept small and consist of reviewers that are “customers” for the product
- Inspections are frequent

Personal Defect Management

- The most economical and effective time to remove defects is when they are injected.
  - The engineers will best recognize them.
  - They will understand what the program was supposed to do.
  - They are most likely to make a correct fix.
- The minimum fix time is when engineers
  - fix their own defects
  - make the fix shortly after the defect was injected
- Checklist based reviews and inspections are effective because defects injected by a particular person tend to be repetitive
- An unexpectedly high defect density in compile or test indicates and ineffective review - product should be re-reviewed prior to continuing with compilation or test.

Quality Management

Fundamentals

- Integration and system test rework cost 30%-50%
- Defects cost 10x - 15x to correct later in the life cycle
- 80% of defects occur in 20% of the work products
- Defect removal yields are approx. constant for a given, appraisal process
- Reduce integration re-work by preventing bad product from moving forward

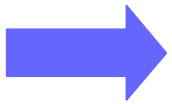
Strategies

- Goal: > 95% yield prior to integration and overall performance improving 15% - 20%
- Technique:
  - distinct design / code activities
  - bench checks of design and code prior to compilation
  - checklist based peer review redundancy control
  - quality indicators as exit criteria
  - re-inspect instead of fix on the fly when module has high defect rate
  - scrap the worst code

Indicators

- Design time >= coding time
- Design review time > 50% design time (matches injection and removal rates)
- Code review time > 50% coding time (matches injection and removal rates)
- Compilation defects < 10/KSLOC (makes probability of secondary injection low)
- Unit test defects < 5/KSLOC (makes probability of secondary injection low)
- Quality Factor > 0.5 correlates with defect free code during integration based on limited SEI data set

Quality Management Plan



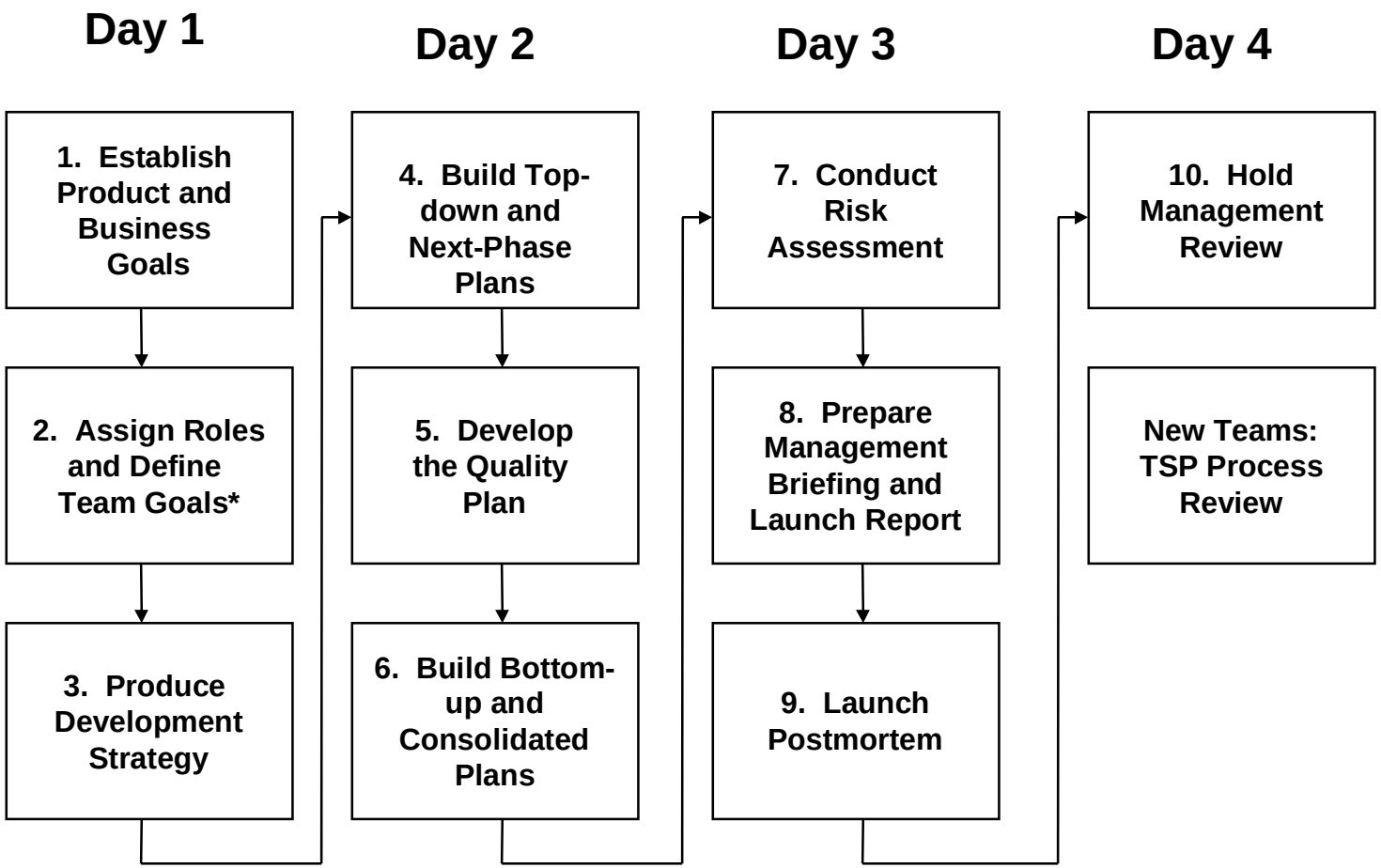
Activity			
Planning (entry criteria)	Development (defect injection)	Appraisal (defect removal)	Postmortem (exit criteria)

	Defects leaked from prev phase	New Defects Injected	Phase Yield	Defects Contained	Defects Leaked
Design	0.00	40	0%	0	40
Personal Design Review	40.00	0	70%	28.0	12.0
Team Design Inspection	12.00	0	70%	8.4	3.6
Code	3.60	60	0%	0.0	63.6
Personal Code Review	63.60	0	70%	44.5	19.1
Compile	19.08	0	50%	9.5	9.5
Team Code Inspection	9.54	0	70%	6.7	2.9
Unit Test	2.86	0	50%	1.4	1.4
Integration Test	1.43	0	50%	0.7	0.7
System Test	0.72	0	50%	0.4	0.4
CUSTOMER	0.36				

TSP

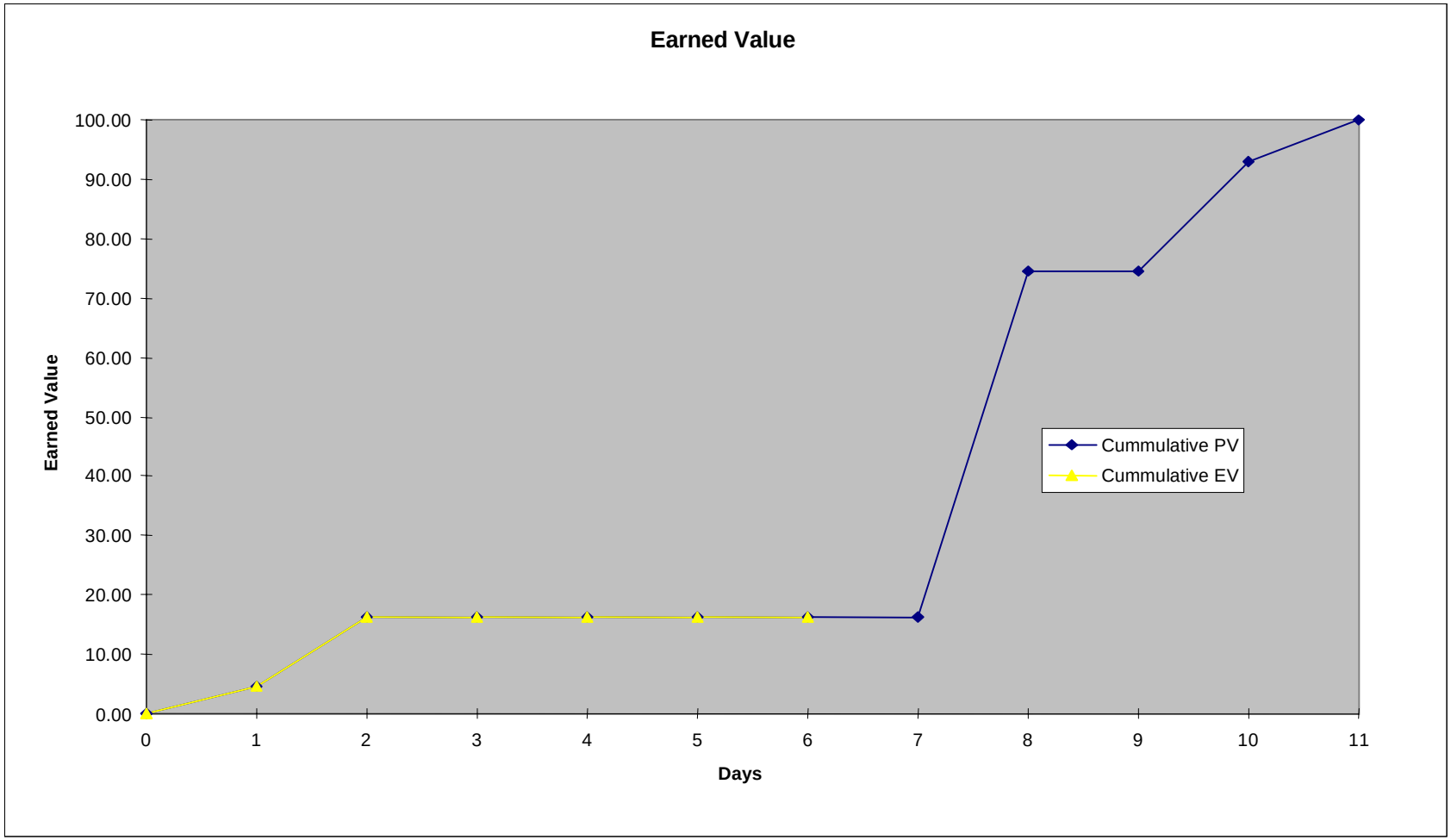
- Team oriented approach to project planning and tracking
- Launch meeting kicks off a project
  - Three to four day process
  - Team building experience
  - Each day, the team works until the day's agenda is complete
  - Generates top-level program plan and detailed plan covering the next three months
  - Data driven, emphasizes individual ownership, focuses on attaining overarching business goals
  - Detailed plan features 0-100 milestones with 2 /person/week granularity that provides EV tracking with extraordinary fidelity
  - Everyone comes out with a clear understanding of roles, responsibilities, and tasks for the next three months
  - Detailed quality plan that projects defects injected and removed by phase, establishes phase exit criteria, defines corrective action plans
- Structured weekly status meeting are used to manage the project
  - Each person brief performance relative plan, risk status, action item status, role related activities etc.
  - Lead briefs overall status, sets goals for next week
  - Lead pro-actively manages to plan
- Quarterly re-launches create new detailed plans
  - Due to extraordinary high level of task granularity, detailed planning is only done one quarter out

Launch Agenda



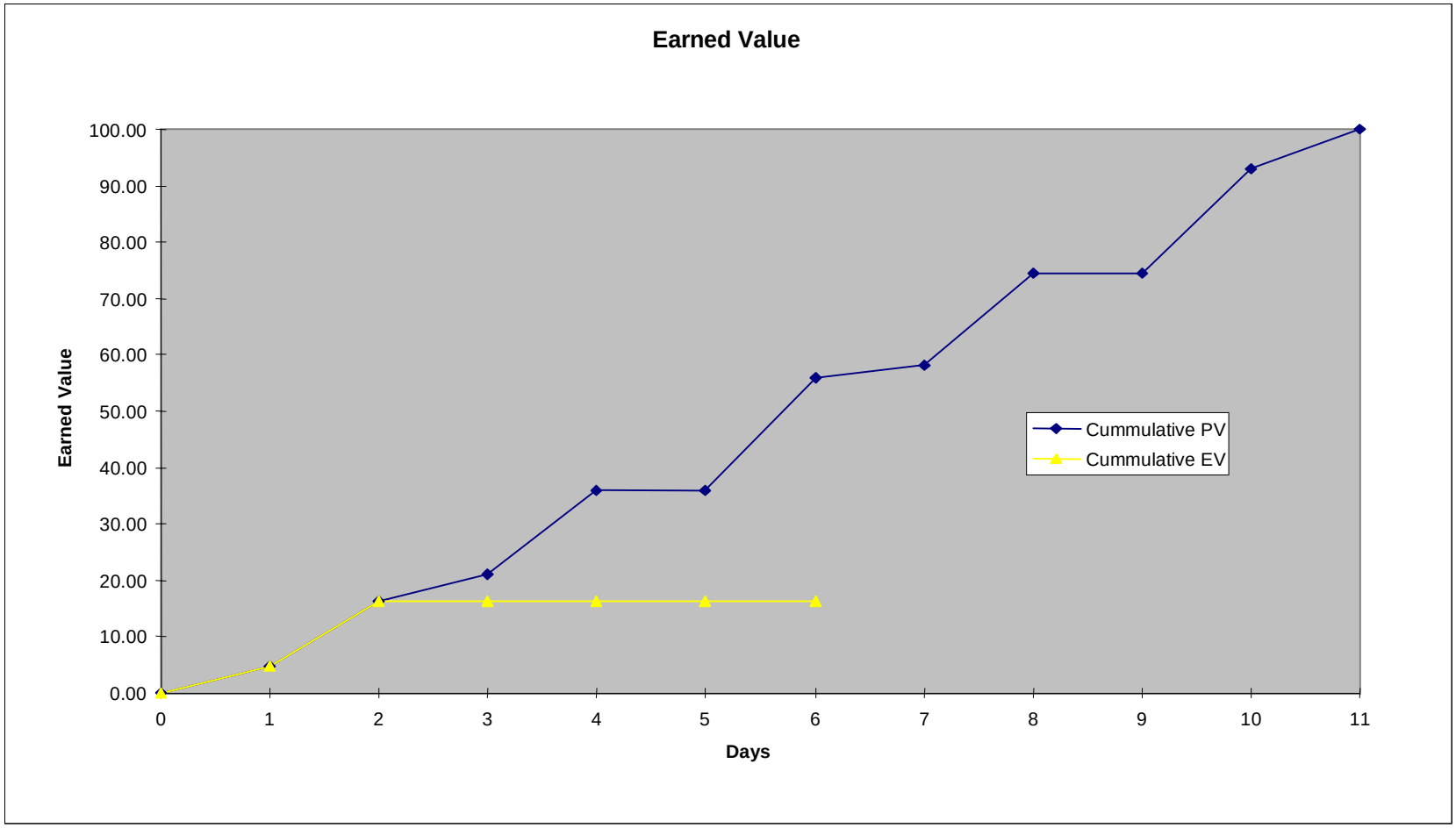
*Team assigns roles. Each role is a focal point for a "management" activity that would ordinarily be performed by the group lead. Distributes responsibility and avoids bottlenecks

Will this activity complete on time?



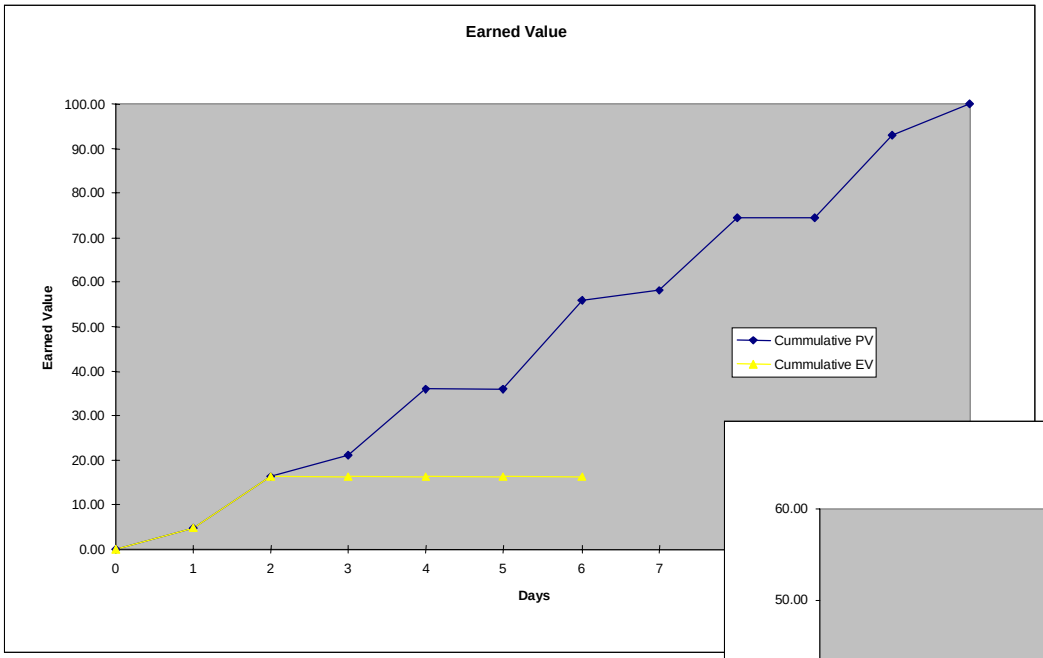
We can’t tell due to the relatively coarse plan granularity!

Can we tell now?



Can we tell why there is no progress?

Why are we behind plan?

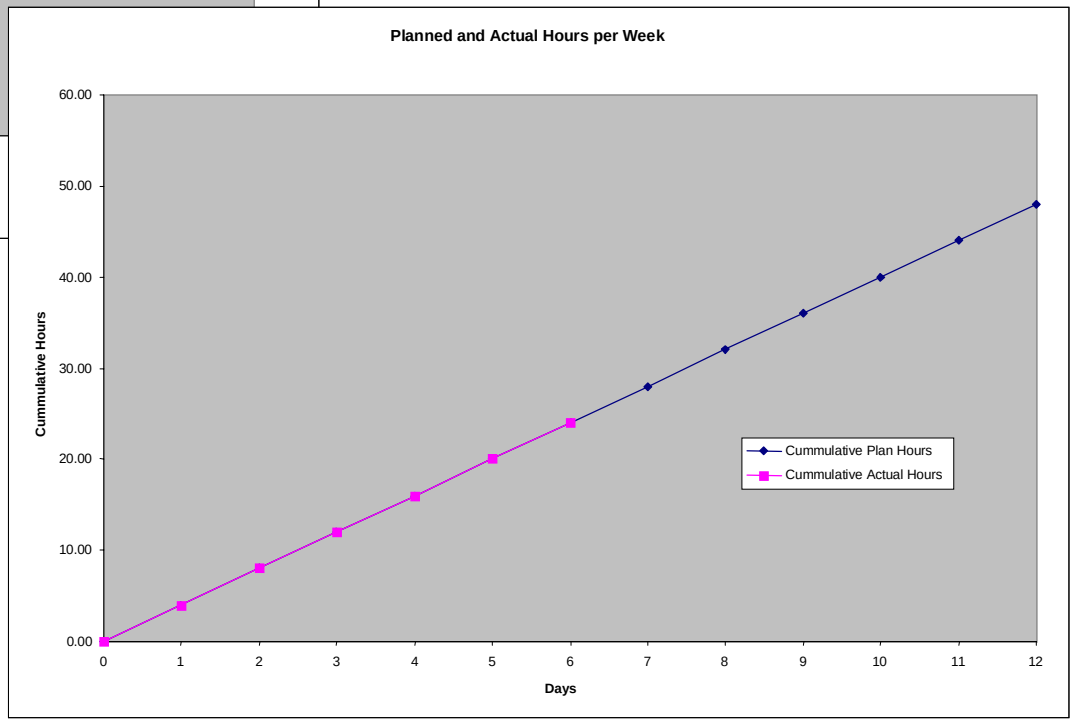


EV behind plan

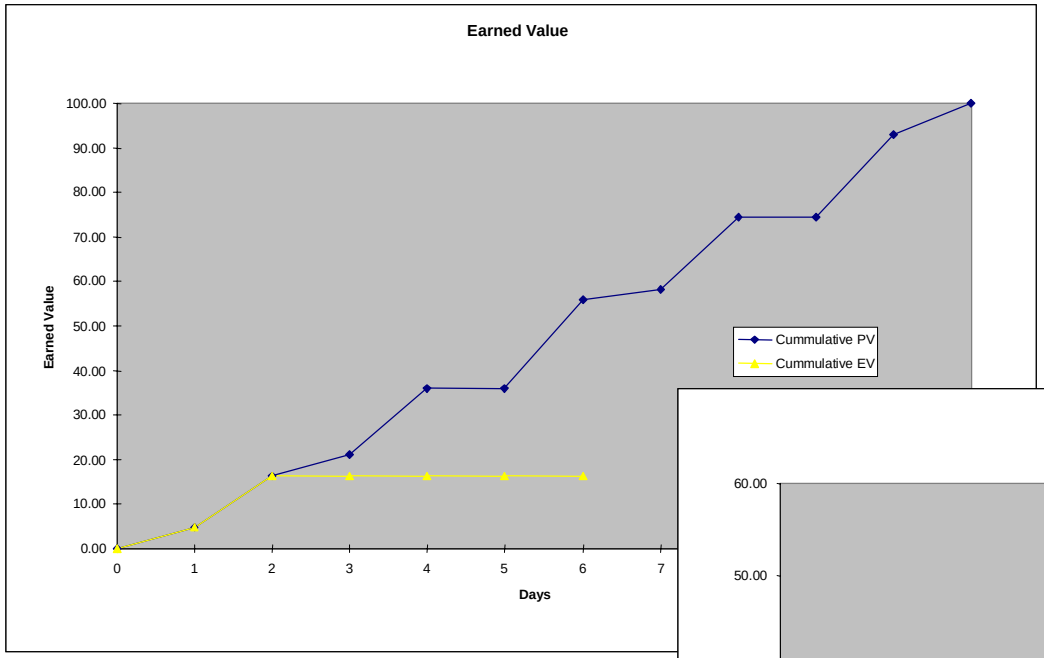
Task is larger than estimated

Or waiting on a dependency

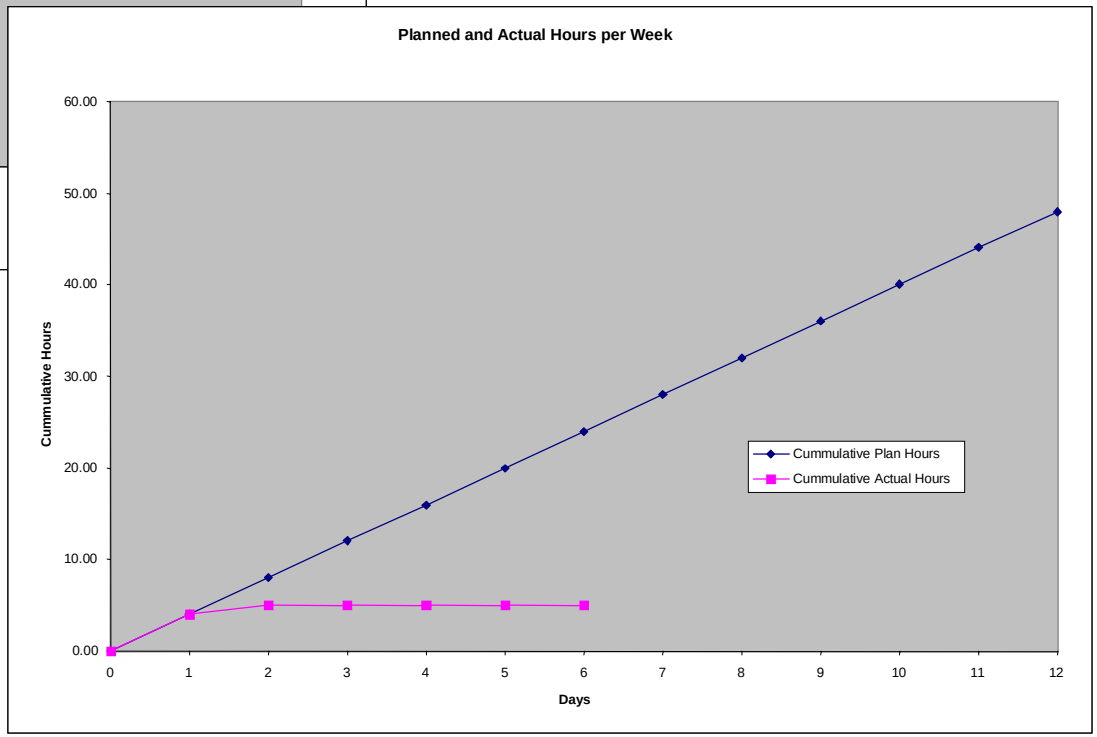
Task hours on plan



Why are we behind plan?



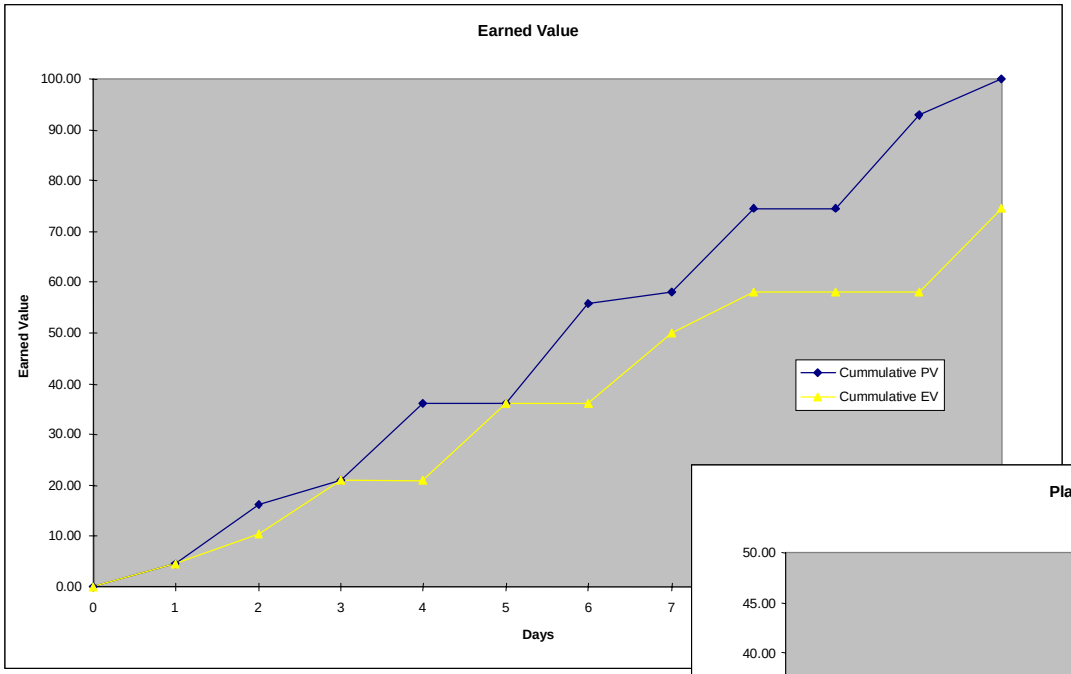
Fix the task hours and EV will come back to plan!



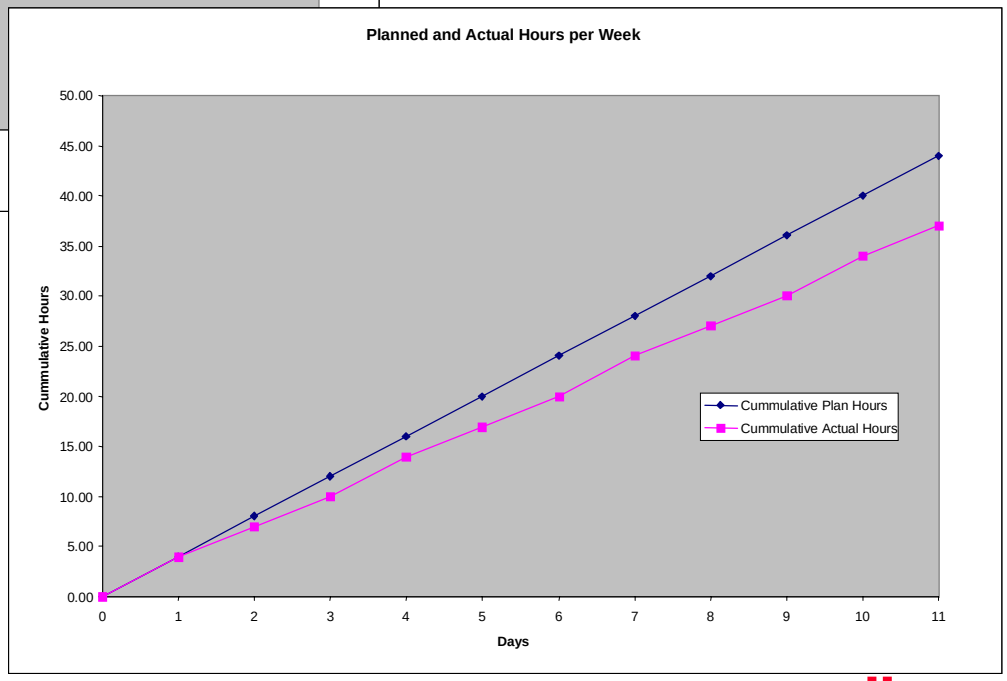
Task hours flat

- Absent
- Working something else

Why are we behind plan?



Pick out the driver and work it!



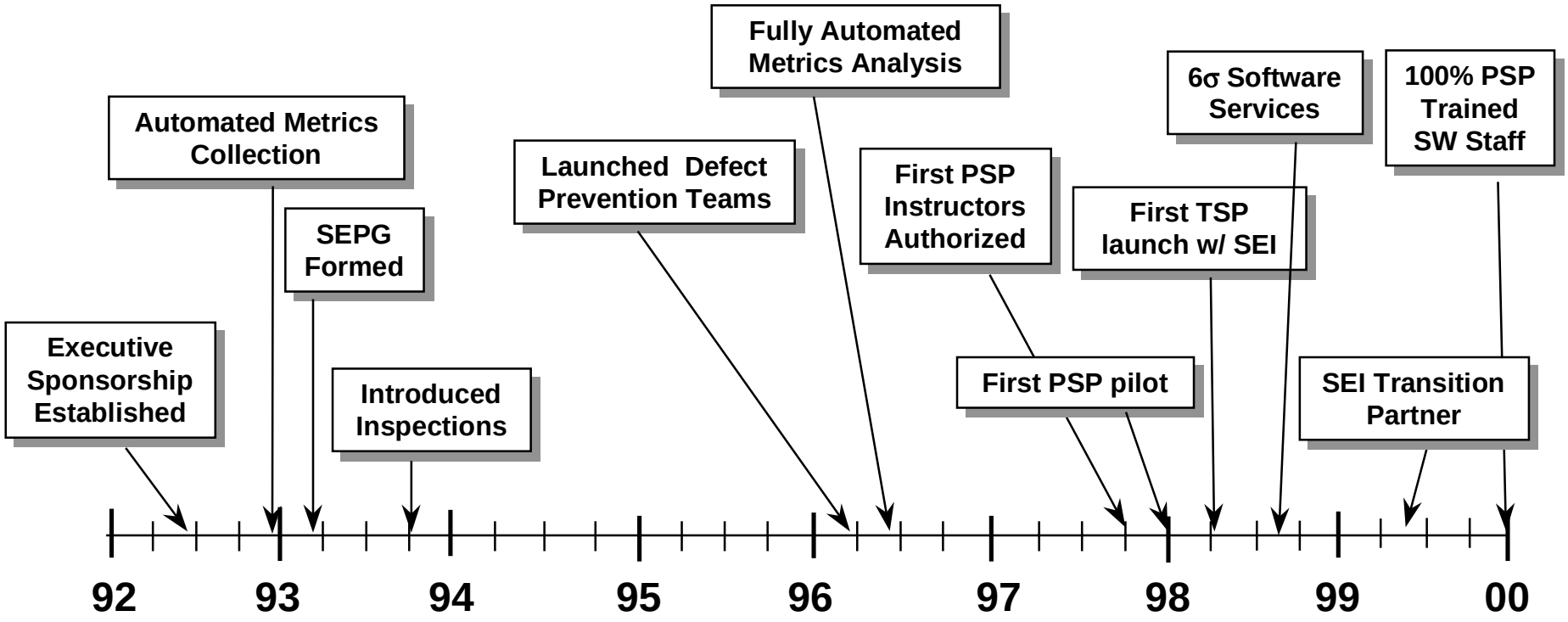
Task hours behind plan

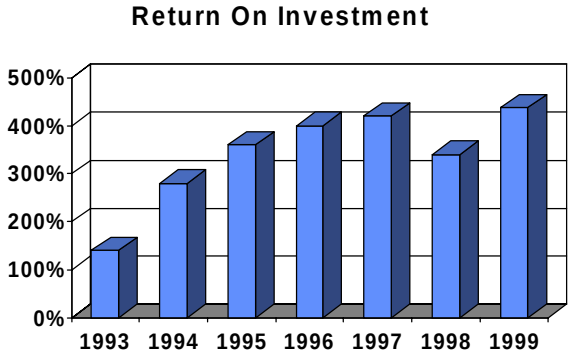
- Accounts for most, not all EV loss

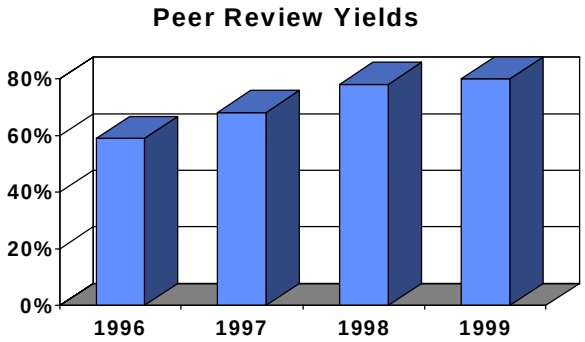
TSP – Bottom Line

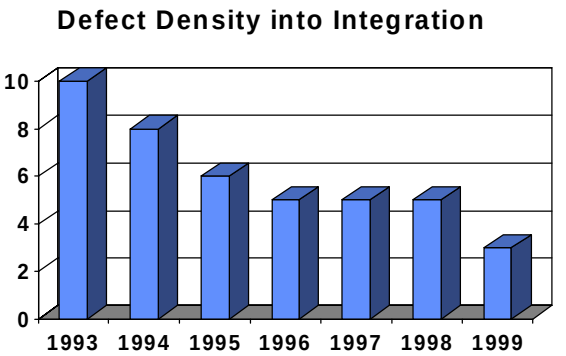
Five people working for 4 days will generate a far higher fidelity plan than one person working alone for 20 days.
They will do it faster than single person.
They will own it.
They will use it.

Experience – Teterboro









Honeywell

Experience to Date

- Technical Training
  - 130 executives
  - 225 managers
  - 250 engineers
- 12 organizations (Honeywell & external)
- 21 launches
- 2 management launches
- 10 pilots (in progress or scheduled)
- Change Agent Training
  - Change management
  - Personal skills
  - Consulting skills

Large Avionics Project Pilot Data

- Introduced on last cycle of embedded avionics program
  - Software staff approximately 30
  - Program has a history of missed commitments
  - About half complete at the time
- PSP used for the last build cycle
  - Overall estimating accuracy 7% low (27 weeks planned vs. 29 weeks actual)
  - Reduction in defect escapes into integration & test over pervious cycle > 4x
- Other data
  - Attrition rate 3% vs. site average 15%
  - Program manager stated: “I never missed a significant milestone once PSP was deployed.”

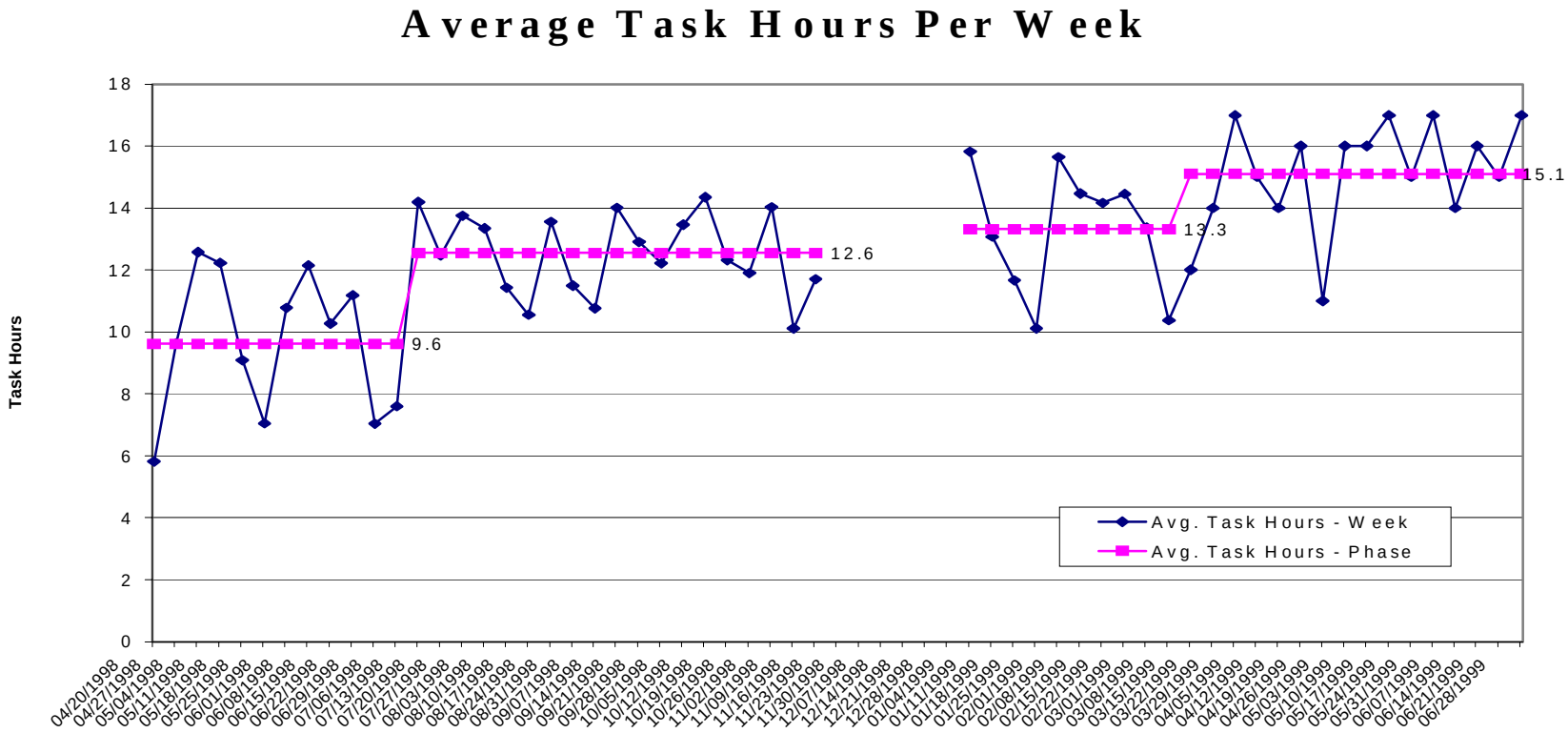
Flight Control PSP/TSP Pilots

- Two small scale PSP/TSP pilot programs involving flight control upgrades
- Started in July 00, currently in progress.
- PSP/TSP process applicable to software and systems engineering activities
- Initial results
  - 4500 new and modified lines of code complete and delivered
  - Under-ran initial estimate by 20%
- System Test engineers on the project state that they are spending their time confirming functionality rather than tracking down bugs, compared to projects prior to PSP introduction
  - Insufficient data from past projects to quantify savings
- Increased time on task per week
  - From 8 to 13 hour per week, a 1.5x improvement
- Reduced cost of quality
  - From 30% to 20% of total task time, a 33% improvement
- Increased productivity
  - 2.0x improvement
- Training time already recovered

Brokering Platform

- Part of large Java based system developed by a financial services company
- Worked jointly with SEI to pilot PSP and TSP multi-team
- Delivered 1 month, (12.5%) late on an 8 month schedule
- 9000 new and modified lines of code delivered
- Team achieved zero defects in certification

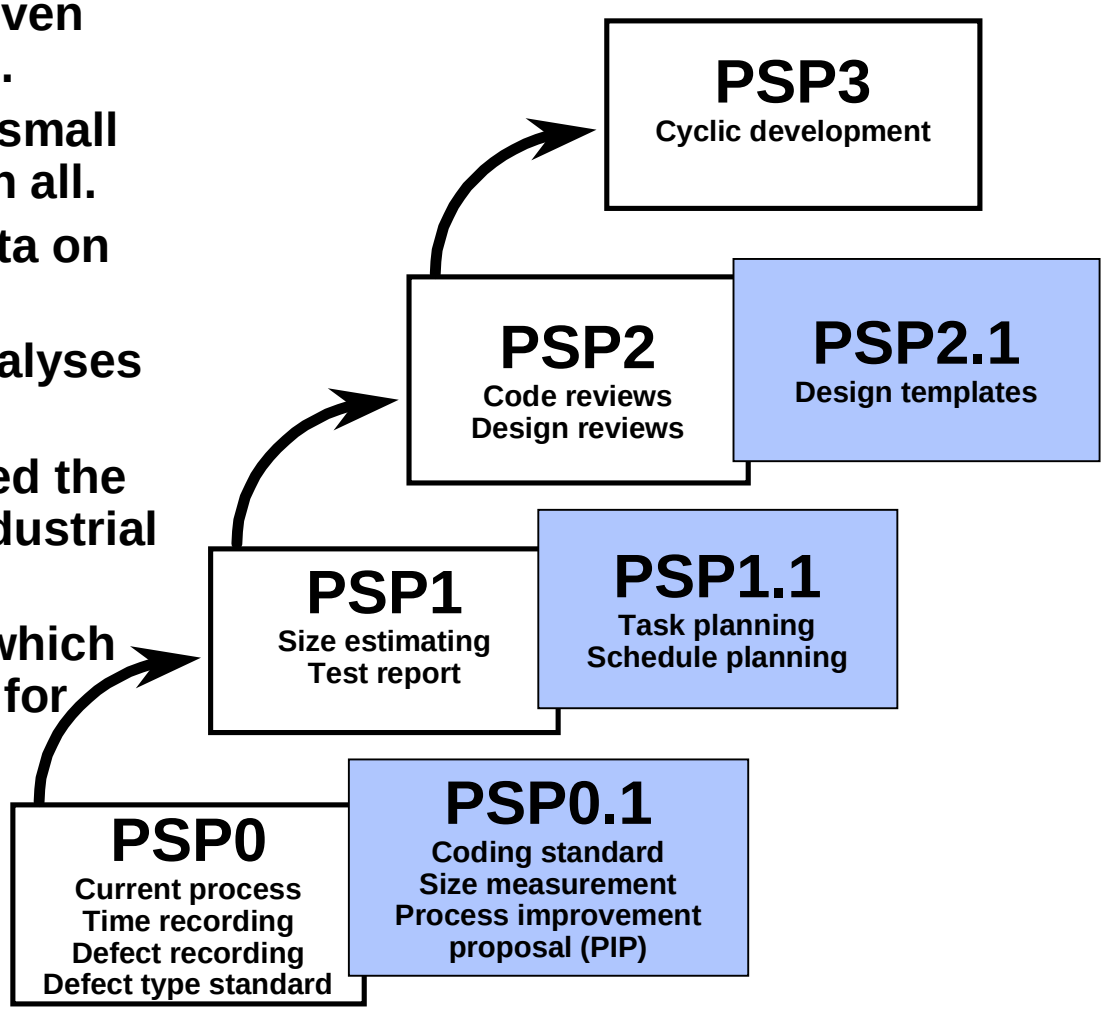
Pilot Task Hours Run Chart



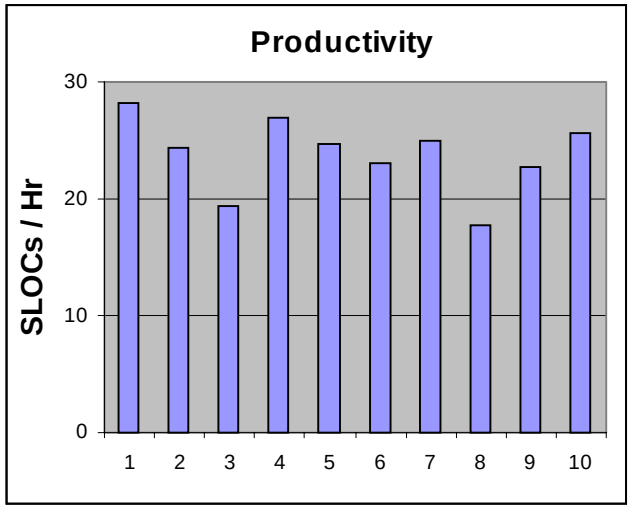
- **Run charts used for task time management and EV analysis**
- **Initially averaging less than 10 task hours/week**
- **Shifted to 15.1 task hours/week (due to quiet times, better documentation, fewer and more efficient meetings, etc.)**
- **Eventually reached 18 task hours/week - a direct productivity improvement**

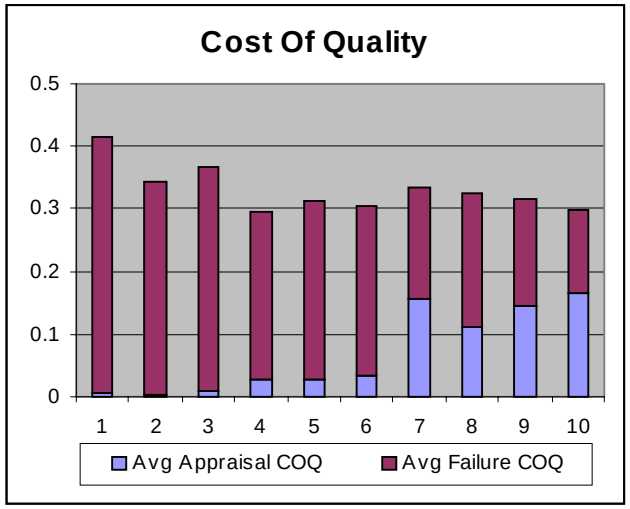
PSP Training

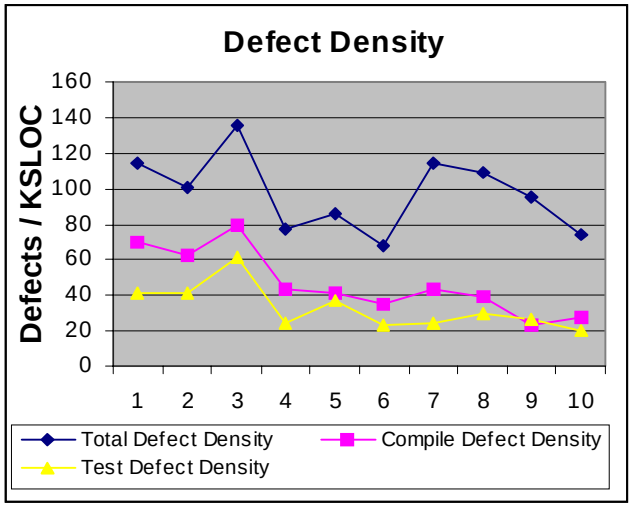
- The PSP is introduced in seven upwardly- compatible steps.
- Engineers write one or two small programs at each step, 10 in all.
- They gather and analyze data on their work.
- They use these data and analyses to improve their work.
- Engineers will have practiced the key elements of a level 5 industrial process.
- Engineers will understand which methods are most effective for them.
- They will do better work.
- They will have long-term improvement goals.



PSP Training Data





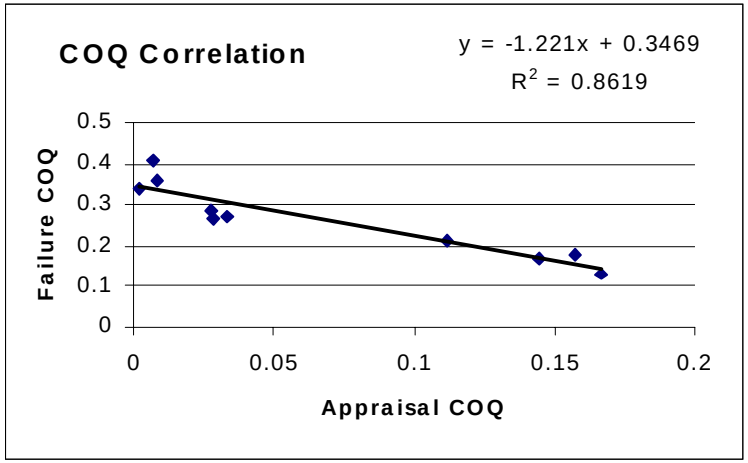


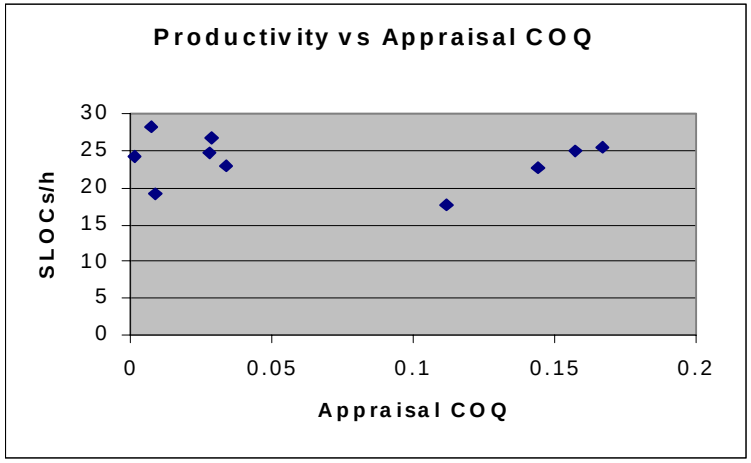
PSP	Productivity (SLOCs/hr)	COQ	Total Defects/KSLOC	Test Defects/KSLOC	Avg Defect Fix Times (minutes)
1.x	24 ± 3.1	33.9 ± 4.5	96.9 ± 23.8	48 ± 11.2	8.9
2.x	23 ± 3.6	31.9 ± 1.5	98.2 ± 17.7	25 ± 4.3	5.4

- **Teterboro training data, PSP 1.x is a level 2 process, PSP 2.x is a level 3-5 process**
- **Avg. COQ remains flat, variability drops by a factor of 3 increasing predictability**
- **Avg. defect fix time decreases by 40% per defect due to extensive use of reviews**
- **Test defects drop by factor of 2, quality of product entering integration at least doubles, resulting in expected integration improvement of 50%**

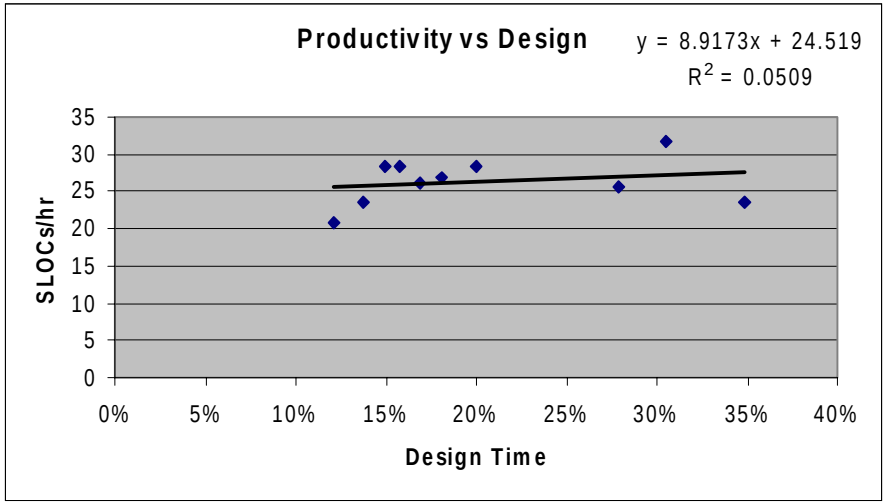
Quality Is Free

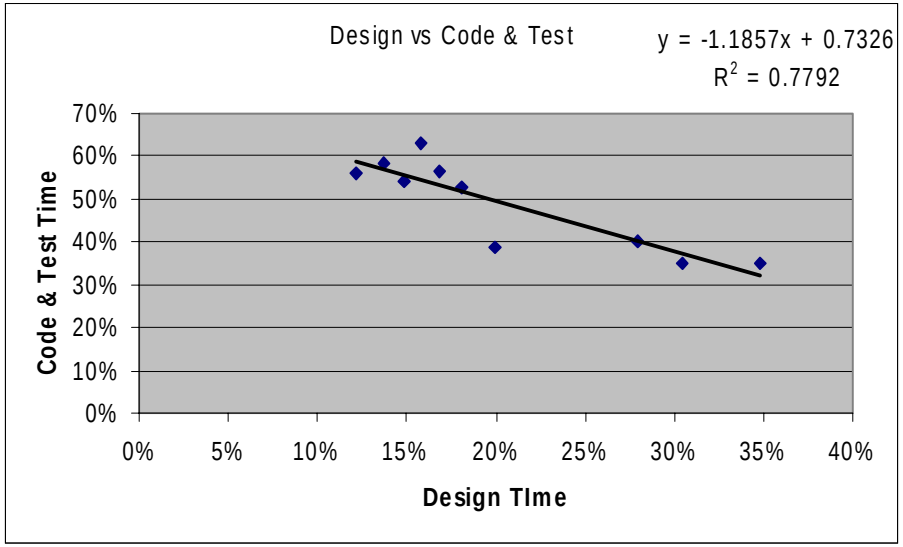
- **Improved predictability: Average COQ remains flat, but variability drops by a factor of 3**
- **Total defect density stable: 100 defects/KLOC**
- *Since test defect density drops by a factor of 2, quality of product entering integration at least doubles, resulting in an expected decrease of integration effort by 50%*
- **Decrease in average fix time (from 8.9 to 5.4 minutes per defect) due to extensive use of reviews**
- **Linear correlation between appraisal time and decrease in failure time**
- **No correlation of increased appraisal time with productivity**





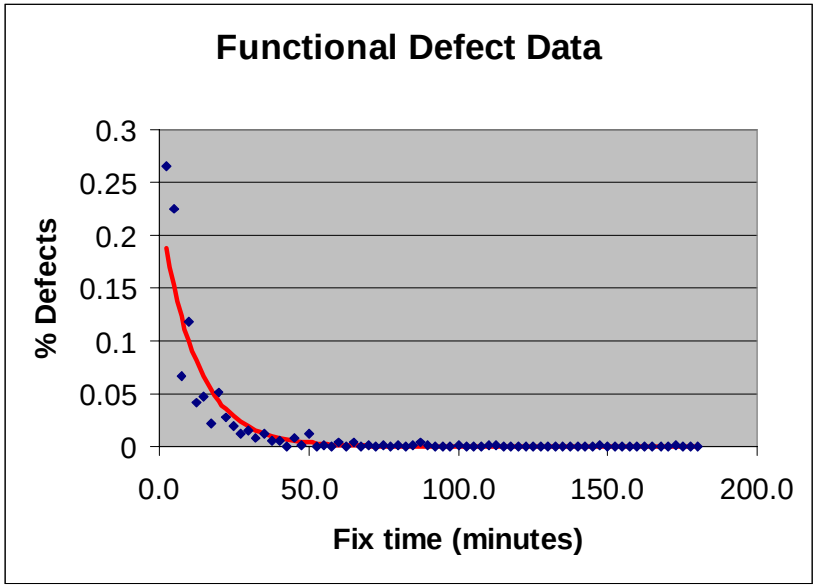
Design Is Also Free





- **No significant correlation between productivity and fraction of time spent in design**
  - **producing formal designs does not lower productivity**
- **There is a correlation between increase in design time and decrease in code & test time**
  - **each 1% increase in design time correlating with a 1.19% decrease in the time spent in code & test**
- **This indicates a direct decrease in coding time when a more complete design is available coupled with a decrease in test time owing to a better review process based on distinct design artifacts**

Defect Statistics



Phase	Avg Fix Time	Max Fix Time
Architecture review	2	2
Design	2.50	5
Design review	3.39	31
Code	3.50	19
Code review	1.97	20
Compile	2.84	147
Test	13.33	185
Escapes	8.25	45

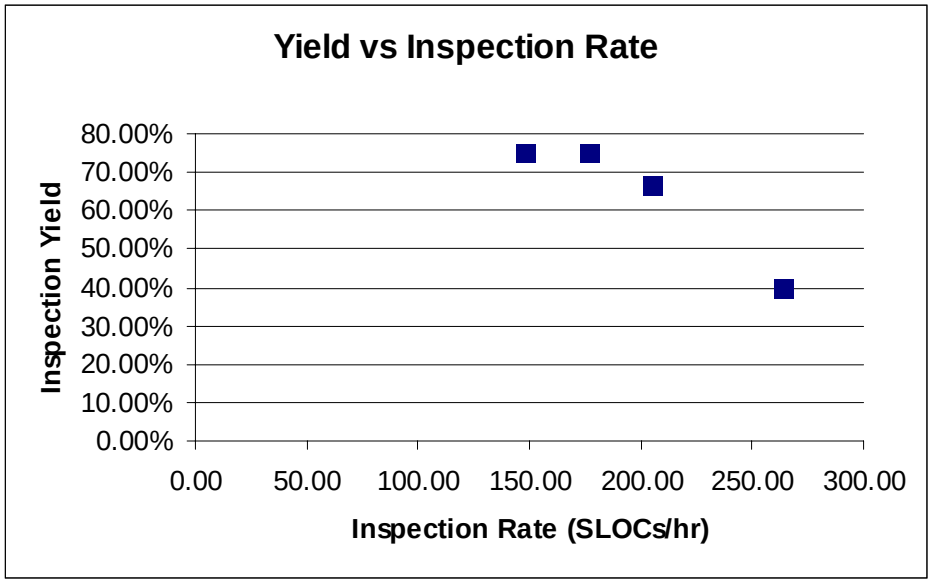
- **Defects are statistically predictable**
- **Reviews are highly leveraged relative to unit test and even have some leverage relative to compilation in a typical PC-based visual environment**

PSP Leakage Matrix

	Design Review		Code Review		Compile		Test	
	1a-6a	7a-10a	1a-6a	7a-10a	1a-6a	7a-10a	1a-6a	7a-10a
Design		2-80-6					6-80-10.2	2-80-11.0
Code				2-20-1.0	3-20-2.0	1-20-1.0	2-10-5.5	1-40-1
				7-40-1.1	4-40-1.25	4-40-2.2	1-20-15.0	
				2-50-1	3-50-1.67		2-50-32.5	
				1-80-3			6-80-13.2	
Compile						1-100-2		
Test							1-20-1.0	
							1-80-5.0	
							2-100-5	

- Cell entries give number of defects-type-average fix time, so that 2-80-6 means 2 type 80's at an average fix time of 6 min each
- Introduction of design reviews reduced the number of design defects caught in test by 50% and decreased their average fix time by about 5 min
- Introduction of code reviews did not change errors removed in compilation significantly, although it did eliminate the type 50 class.

PSP Yield Management



- Personal yield curves can be developed quickly from the first few weeks of data and used to manage appraisal processes
- If the inspection rates go too high, the reviews are not worth doing, to low and their cost can exceed testing
- Similar results apply to group inspections

Getting Started with PSP

- Understanding the cost/benefit relationships
- Building sponsorship
- Selecting a pilot project
- Automation
- Training the engineers
- The TSP launch
- Support structure

PSP/TSP Objectives

- **Pilot Program Objectives:**
  - **Run 6 PSP/TSP Pilots over 18-24 Months**
  - **Reduce integration and test defects by 30%;**
  - **Increase weekly hours on task by 20%**
  - **Break even on a 12 month project**

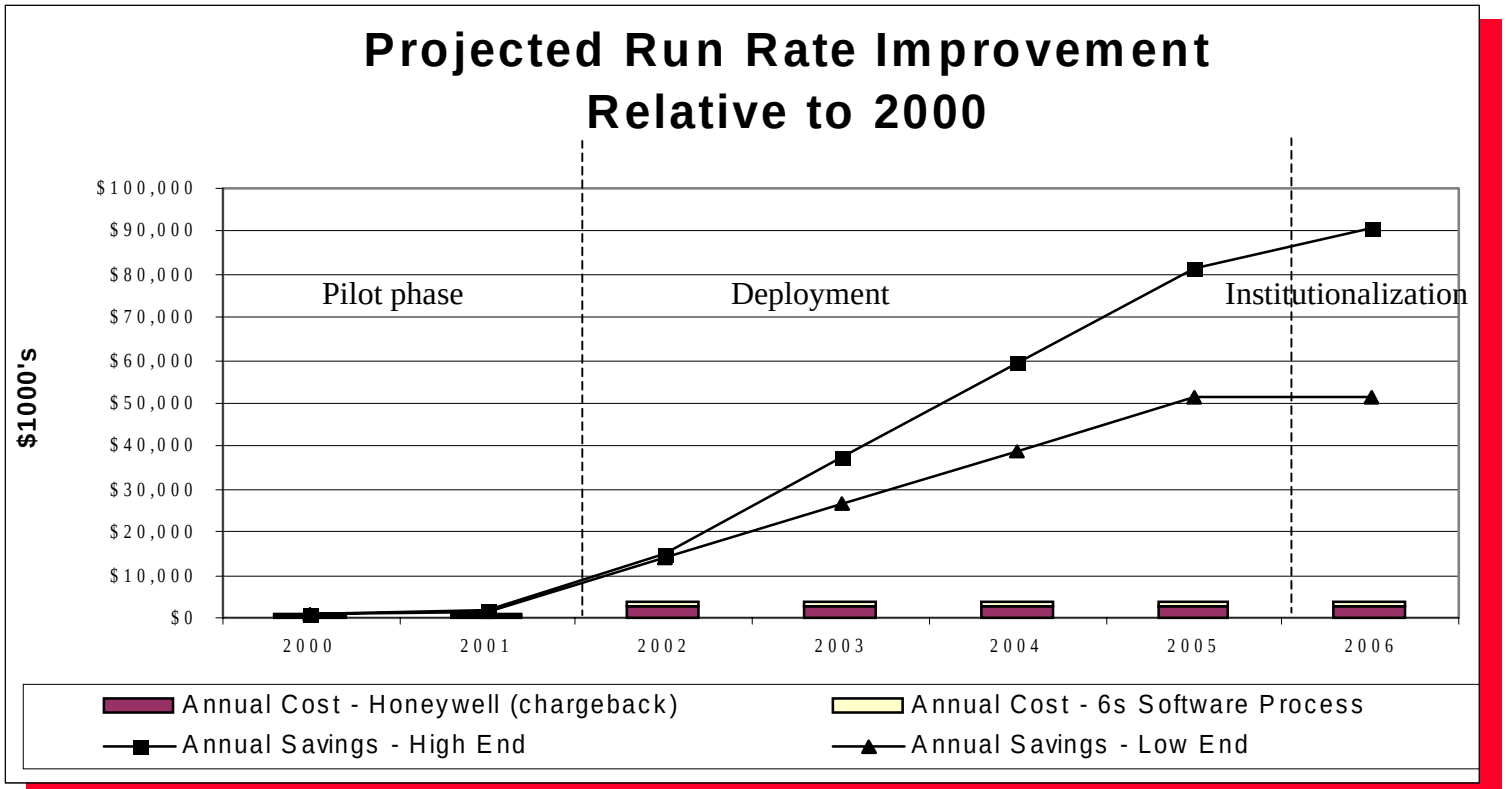
- **Institutionalization Objectives:**
  - **Institutionalize within 4 years after pilot phase**
  - **Fully exploit design for 6σ techniques**
  - **Reduce integration and test defects by 50%; Increase weekly hours on task by 50%**
  - **Permanently reduce software development cost by 25% - 40%**

PSP/TSP Cost Benefit Summary

	Pilot	Institutionalization
Baseline project cost	$980K	$980K
PSP/TSP deployment cost	$202K	$18K
PSP/TSP savings	$249K	$249K – $442K
Quality – integration & test	$103K	$103K – $172K
Schedule – task hours	$146K	$146K – $290K
Net Cost with PSP/TSP	$933K	$556K – $749K
Savings	$47K	$231K - $424K
Run rate improvement	5%	23% - 43%

Post-institutional savings range goes from a low-end comparable to pilot project to a high-end that assumes a greater level of process maturity and use of Six Sigma methods that might not be possible with a less capable process

Projected Organizational PSP/TSP Savings



- **25% to 40% cost reduction at full institutionalization**
- **annual cost avoidance of $50 to $90M**

...the power of quality on the scale of Honeywell's workforce

Building Sponsorship

- Identify a local champion
  - typically someone active in software development or process improvement that has been exposed to PSP/TSP at a conference
  - needs to have a position of influence in the organization
  - needs to be interested in trying it out
- Establish linkage of PSP/TSP to organizational initiatives
- Provide senior management with “PSP Executive Seminar” training
- Provide “Managing PSP-trained Engineers” training to first and second line supervisors and managers
  - a 2-day version of the course to make it accessible to managers with busy schedules
  - “PSP Executive Seminar” does not provide adequate technical depth for this management group
  - Instructors with first-hand PSP/TSP experience provide credibility for managers and executives
- Solicit volunteers for pilot projects and perform cost/benefit analyses

Selecting a Pilot Project

- Lower maturity level organizations can implement PSP.
  - need basic CM and QA in place
  - desirable to have a documented process capability baseline
- Pilot project should break even within one year.
- The first line supervisor “sells” PSP to the pilot team.
- A PSP-trained team mentor should be selected from outside the project.
- In selecting the initial projects, try to pick ones that are not in perpetual crisis.
- In selecting the initial team, try to pick members who are open-minded.

Training the Engineers

- Conduct a change management workshop with first line supervisors
- Brief the engineers on the PSP/TSP process and the reasons for selecting the pilot
- Provide just-in-time training prior to project start
  - One week on, two weeks off, one week on
  - Dedicated training facility; preferably offsite
  - Each student is provided with a laptop computer
  - Automated tool use during training to facilitate data capture and analysis
  - Entire team is trained together
- First line supervisor and mentor trained with the team *after* completing the “Managing PSP-trained Engineers” class
- Reward developers for taking the training and for completing the homework assignments

Automation

- Robust automation essential for metrics collection and analysis when using PSP/TSP
- Provided by ISPE, a multi-user client-server database application that includes support for:
  - scheduling meetings and inspections and maintaining meeting and inspection records
  - logging and tracking the status of action items
  - risk management
  - problem reports
  - TSP launch planning and tracking
  - Quality Plans
  - Earned Value
  - TSP status meetings
  - PSP time and defect logging
  - estimation (PROBE+)
  - comprehensive automated metrics collection and analysis
- Provides dynamic views for real time data filtering and aggregation
- Low data collection overhead - 12 sec for a time log entry, 30 for a defect log entry, several minutes to produce a typical estimate
- Data privacy and security

Support Structure

- Senior management support
  - Vertical alignment of goals is essential. Each level of the organization must see a clear benefit from practicing the new process
  - Sustained demonstration of management interest is essential – regularly ask for PSP team metrics
- Team mentor
  - Follow-up mentoring is essential.
    - It is much easier to get the team to collect data than to use it effectively
    - The mentor needs ensure consistency and to help team members with data analysis during post-mortem
  - Teams take 3 – 6 months to become comfortable with applying the full process on a real project
- Rewards and recognition
  - reward pilot team for taking the class and for completing the homework assignments

Lessons Learned

- Use of TSP is essential
- Robust automation is a pre-requisite
- All team members should complete all PSP assignments prior to TSP launch
- Set uniform standards for everybody and for all work products
- Explicitly identify public and private data “up front”
- PSP principles can be applied to other lifecycle phases, provided adequate support and training are provided
- Use of statistical process control techniques is essential for effective task hour management
- Staff design skills emerged as the limiting factor in achieving plan granularity and in increasing task time

In Summary ...

- **PSP/TSP is a real-time closed-loop level 5 process PSP kills variation by eliminating the overriding causes of variation - the author and time measurement errors**
  - **If this source of variation is not eliminated, you will not get a stable process or usefully narrow process limits for anything except inspections**
  - **once the variation is removed, the software process becomes surprisingly predictable in terms of rules of thumb and "magic numbers"**
- **There is no trade off between quality and cost because there is a cost benefit not a cost penalty when you produce software that is approximately defect free**