



# *Are Your Software Estimates Baseless?*

*Jairus Hihn (312)*

*Marti DeMore (318)*

Jet Propulsion Laboratory,  
California Institute of Technology

*7/25/2012*



# Disciplined Processes Make A Difference

The data shows that projects that are following a disciplined process exhibit

- ✦ Below average budget growth (Historically the average was 55%)
- ✦ Higher productivity
- ✦ Lower defect rates

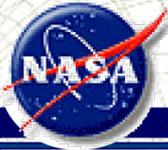
Flight Software Key Process and Product Metrics			
Process Performance	Effort Growth from PDR	Productivity (Lines of Code/ Work Month)	Defect Density (Defects/ Thousand Lines of Code)
Robust Process	39%	150	4.3
Low to Moderate Process Performance	116%	106	5.9

While not all the difference is due to following a disciplined process, it is clearly a contributing factor.



# *Basis of Estimate*

- ✦ A key element of a solid planning methodology and process is having a robust Basis of Estimate (BOE)
  - ✓ Fundamental to developing a defensible estimate
- ✦ A Robust BOE
  - ✦ Provides a strong baseline for evaluating the cost of the inevitable changes that come on a SW task
  - ✦ Provides detailed information that can be used to identify the causes of budget and schedule overruns/underruns
  - ✦ Provides a foundation for improved cost estimates in subsequent years or projects, when you monitor actuals
  - ✦ Data-based negotiation is more effective than opinion-based negotiation.
    - ✦ **“When I bring data I win my arguments”**



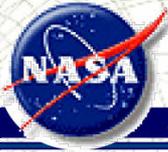
# What is a BOE?

- ✦ **The term ‘Basis of Estimate’ or BOE is used in various ways**
  - ✦ We all know what we mean but do we really mean the same thing?
  - ✦ It is highly unlikely
- ✦ **Most often the focus is on the documentation**
  - ✦ But it is really about the work you do in developing a robust estimate
- ✦ **JPL Costing and Pricing Office**
  - A defensible BOE embodies the following traits:
    - Transparency – The BOE must document and clearly show the rationale used to derive the cost estimate
    - Consistency – Information referenced in the BOE should match the source information, be it in the technical proposal/descriptions, historical documents and/or other parts of the effort
    - Accuracy – Calculations must be documented and correct
    - Due Diligence – Does the documented estimating process represent the best available cost estimating process and data?



# MSL Lessons Learned: BOE

- ✦ It was recognized that a contributor to the excessive cost growth observed on MSL was not fully understanding and not communicating the development cost
- ✦ Key recommendations
  - ✦ Need comprehensive and robust BOE's
  - ✦ Each Section should develop a template to support planning and negotiation of a project or proposal resource plan
    - ✦ Use at major reviews
    - ✦ Ensures that all of the elements of a good BOE are addressed
  - ✦ Capture in WA Tool
- ✦ Several years later SQI captured a software version of many of these recommendations in the Software Product Estimation Process



# *A Comprehensive SW BOE Should Contain ...*

- **Statement of Work and Scope**
- **WBS with dictionary (product oriented work breakdown structure)**
- **Planning Parameters or supporting lower level estimates**
  - **E.g. Software Size Estimates**
- **Supporting Model Estimates and Analogies**
- **Schedule**
- **Effort Estimates with supporting assumptions and detailed BOE**
- **Procurements**
- **Acquisition Approach (If applicable)**
- **Cost estimates**
- **Key Assumptions and Risks**
  - **Significant Cost and Risk drivers**
  - **Risk List/Issues/Known Liens**

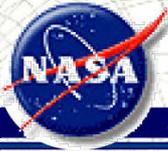


# The JPL Work Agreement Tool - 1

## JPL Work Agreement (WA)

The WA Tool has places to put all of these information

		Identification
1. Customer	Created: 7/23/12 Updated: 7/23/12	3. Form Id: Version - Baseline / Revision: 1 / A
4a. Work Agreement Title: <a href="#">Software Engineering Initiative</a>		
		Scope
9. Statement of Work		
10. Objectives and Products		
11. Applicable Requirements and Constraints		
13. Product Development/Support Plan		
14. Acquisition Approach		
16. Work Breakdown Structure		



# The JPL Work Agreement Tool -2

## Schedule

### 20. Product Milestones/Receivables/Deliverables

Items	From / To	WBS / ORG. No.	Date
-------	-----------	----------------	------

Comments:

### 21a. Budgeted Schedule

## Resources

### 25a. Budgeted Resources

Basis of Estimate: Budget At Complete: \$0 K

Comments

### 27. Recommended Reserves

Reserves	2012	2013	2014	2015	2016	2017	To Complete	Total

## Basis of Estimate

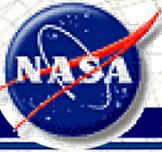
### 30. Basis of Estimate Details



# The JPL SDSP:

## Software Product Estimation Process

- The SPE process is in rules!
- SPE has pointers to templates, examples, tools
- SPE provides high level description of all of the BOE best practices



# How do JPLers Estimate?

**Recently a survey of cost estimation practices was completed across all NASA Centers**

- **60% reported using a WBS**
- **Supporting Model Estimates and Analogies**
  - 41% use more than one method
  - 76% use top level analogies
    - But only 20% capture actuals
  - 59% develop a bottom-up estimate
  - 25% use cost models (Much higher for large flight SW tasks)
- **87% estimate Schedule**
- **Planning Parameters or supporting lower level estimates**
  - 53% size their systems and the majority use SLOC or Modules
- **Risk List/Issues/Known Liens**
  - 33% identify risks
  - 46% incorporate risk and uncertainty into their estimates

# BOE Guidance: WBS



- ✦ A comprehensive product-oriented WBS is one of your most important tools
- ✦ A product-oriented WBS is a planning and reporting tool which outlines the list of work activities to complete the software job by identifying all the elements required to do the software job
  - ✦ The sum of its elements is the entire software job

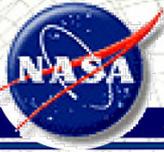


# BOE Guidance: Example Product Oriented Flight Software WBS

Jet Propulsion Laboratory

Software Quality Improvement P

- ✦ Software Management
- ✦ Software Systems Engineering
- ✦ C&DH
- ✦ GNC
- ✦ Payload Accommodation
- ✦ Engineering Applications
- ✦ System Services (includes Fault Protection)
- ✦ Software Development Test Beds
- ✦ Software Integration and Test



# BOE Guidance: WBS

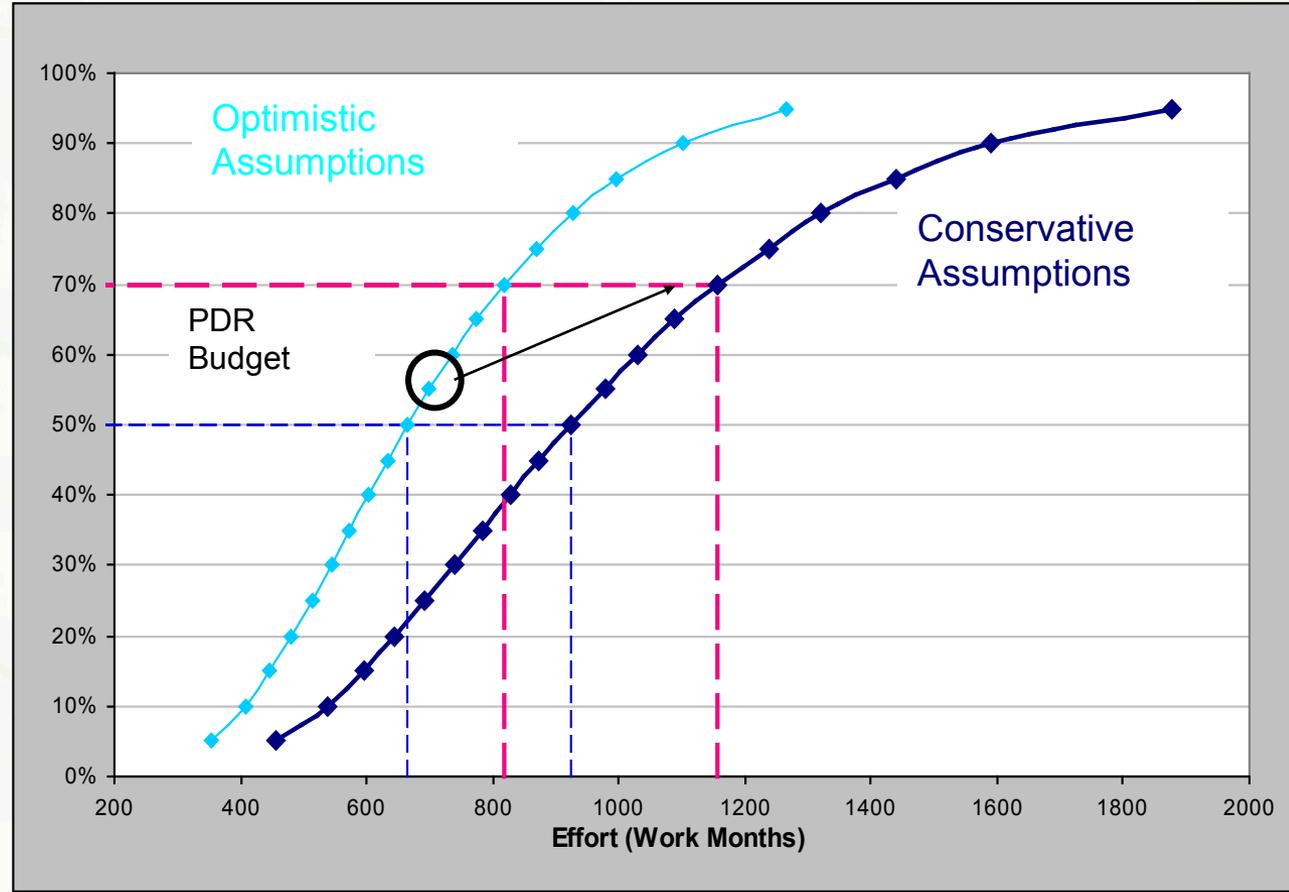
- It is very important to keep your different breakdowns consistent
  - Functional decomposition
  - Cost WBS
  - Schedule elements
- Relationship between these should be identical or easily mappable
- 50% reported developing an integrated technical, cost, and schedule breakdown for the software task or product? .

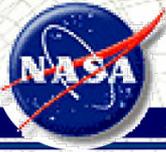


# BOE Guidance: Model Based Estimates

## JPL Software Cost Models

1. SCAT (COCOMO)
2. SEER-SEM
3. Team X FSW Cost Model





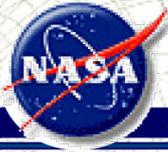
# BOE Guidance: Risk

## Common Causes of Effort Growth

Historically, there is a pattern of being overly optimistic in setting budgets by not taking sufficient account for:

- ✦ Changes and increases in scope
  - ✦ Concurrent hardware development
  - ✦ Inability to scope flight software due to inadequate project definition
  - ✦ Software is used for risk mitigation, but never planned for up front.
  - ✦ Software is the system complexity sponge
- ✦ Testbed and SoftSim availability and maturity
- ✦ Optimistic software inheritance assumptions
- ✦ Anything New
  - ✦ Technology
    - ✦ Autonomy
    - ✦ Precision landing
    - ✦ Hazard avoidance
  - ✦ Design
  - ✦ Language
  - ✦ Tools
  - ✦ Development environment
  - ✦ Processes
  - ✦ Customer or sponsor

***[These items are based upon causes of cost growth observed at JPL.]***



# Resources

- ✦ For assistance accessing and using any of the tools and work aids we mentioned along with many other assets that are available contact
  - ✦ [jairus.m.hihn@jpl.nasa.gov](mailto:jairus.m.hihn@jpl.nasa.gov) 4-1248
  - ✦ [scott.c.morgan@jpl.nasa.gov](mailto:scott.c.morgan@jpl.nasa.gov) 4-4972
  
- ✦ Available SQI resources
  - ✦ Software Product Estimation Process is in Rules!
  - ✦ Code Counters
    - ✦ SLiC and SLiC-Diff
  - ✦ Data available from the SMART Repository
  - ✦ Cost Models
    - ✦ SCAT, SEER-SEM, Flight SW Cost Model
  - ✦ Planning Models (e.g. Percent Effort by Role)
  - ✦ 2011 State of Software Report
  - ✦ Cost Estimation Handbook and Training Materials

# AMPCS BOE Example



- ★ AMPCS is using an innovative approach using Mind Maps to support their design and estimating work

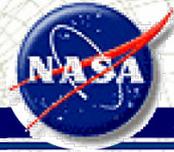


# The Nature of AMPCS Development

Jet Propulsion Laboratory

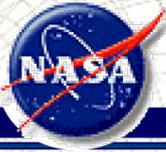
Software Quality Improvement P

- ★ AMMOS Mission Data Processing and Control System is for multimission telemetry processing and flight commanding
  - ★ Currently supporting SMAP, and undergoing port for Odyssey
- ★ All initial code inherited from MPCS for MSL
  - ★ Sparse requirement set and loose processes
  - ★ Characterized by rapid development under pressure, resulting in a highly evolved design
  - ★ Estimates used MBG method (Marti's Best Guess)
- ★ AMPCS difficult to estimate accurately as a result
  - ★ Largely developed under the "Master Programmer" paradigm, with 2 master programmers (one of whom is now gone)
    - ★ No one person has knowledge of the whole design, and it is not consistently written down
  - ★ Good requirements set is only now coming into play
  - ★ Minimal up-front definition of tasks



# An Estimation Quandary

- ★ MGSS processes more rigorous than MSL's and CMMI evaluation approaching; needed better estimation
  - ★ AMPCS schedule still too rigorous to do the more traditional things I knew would help
  - ★ I have only minimal experience in software cost estimation techniques
- ★ For release planning, I desperately needed:
  - ★ A means for gathering requirements and drilling into design for a requested change quickly
  - ★ A way to get all the aspects of a proposed change out of developers heads in order to estimate both development and test impact
  - ★ A means for doing this with a team almost fully occupied with the previous release



# AMPCS Planning Approach

- ✦ Based upon two key elements:
  - ✦ Good use case definitions
  - ✦ Mind Mapping to draw out details from the team
- ✦ “Mind Map” defined:
  - ✦ A diagram used to visually outline information [sic] created around a single word or text, placed in the center, to which associated ideas, words and concepts are added. Major categories radiate from a central node, and lesser categories are sub-branches of larger branches. (Wikipedia)
- ✦ Both elements are interactive and custom to AMPCS
  - ✦ Includes customer, SEs, developers, testers
  - ✦ Mind Map templates are AMPCS-specific; general ones not useful
  - ✦ Use Case Definition template developed by AMPCS, based upon industry approaches



# AMPCS Planning Approach (Cont)

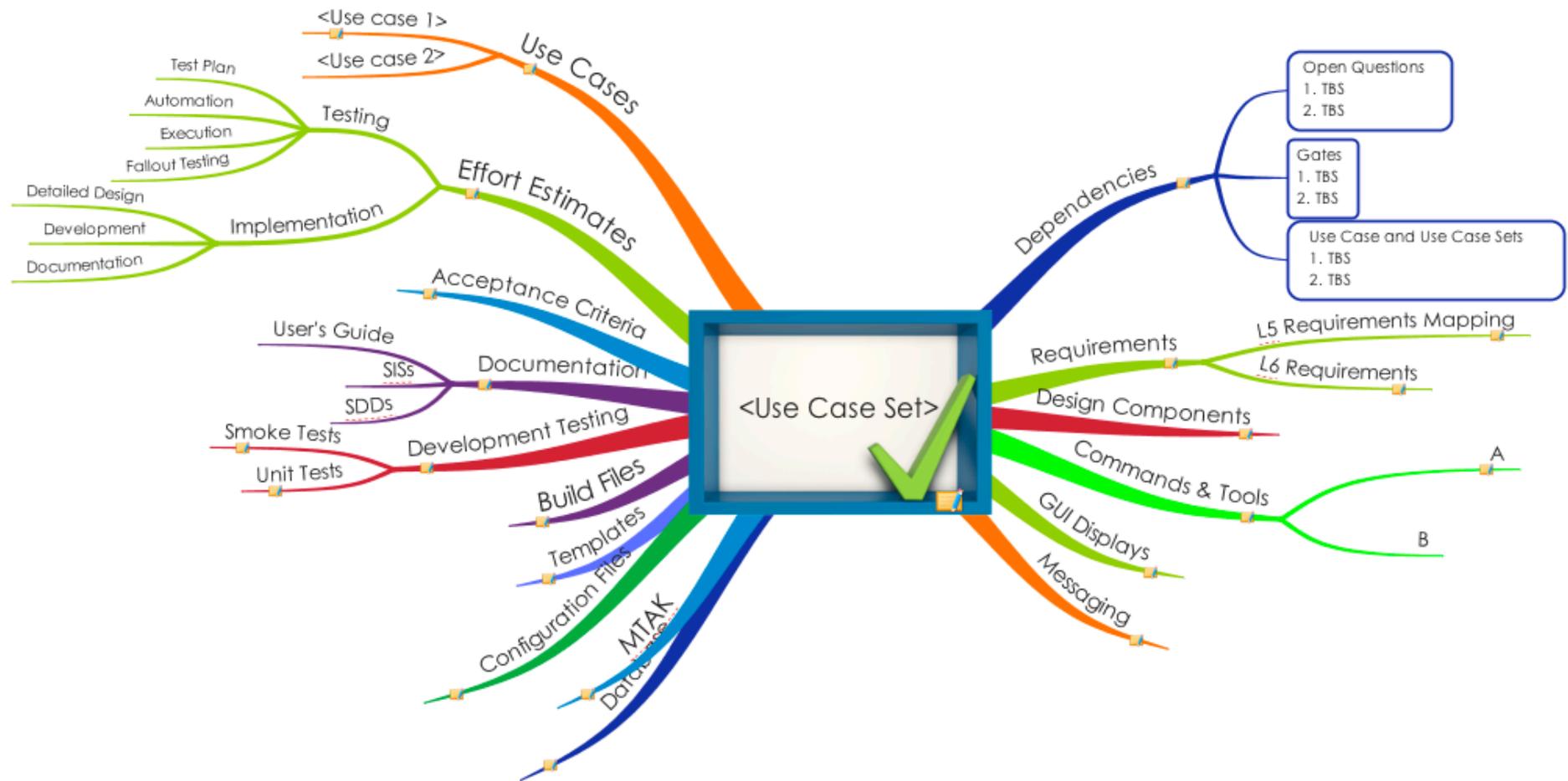
Jet Propulsion Laboratory

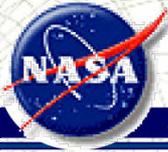
Software Quality Improvement P

- ✦ Estimates computed for Mind Map branches and totaled
- ✦ General process is Use Cases -> Mind Map -> Implementation Plan -> Schedule
- ✦ We currently use the iMindMap software from ThinkBuzan:
  - ✦ It allowed notes to be added and later exported for documentation
  - ✦ It allowed task durations and dependencies to be attached to branches and exported to Microsoft Project
  - ✦ Because we all agreed “it looked cool”



# An AMPCS iMindMap Template





# iMindMap Notes for Export to Word



Open Questions  
1. Does chill\_plot\_drf\_gui really send plots to the printer, or just

Notes

Text Layout Table Insert

Default font

**B** *I* U

This set of tools already exists in the MSL adaptation stream. The following changes are required:

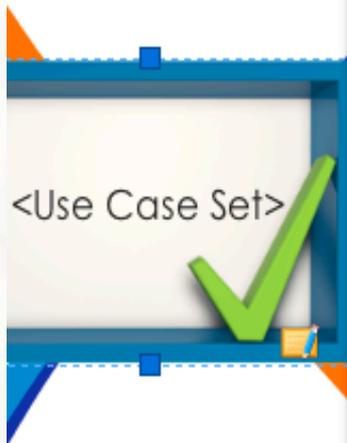
1. Make these tools report standard AMPCS version information.
2. Convert them to use a multimission channel dictionary parser. Currently, the Perl code is hard-coded for the MSL dictionary format.
3. Move the code and any build tasks related to them from the MSL AccuRev stream into the core stream.

The list of affected tools is:

- chill\_get\_drf
- chill\_int\_report
- chill\_int\_report\_viewer
- chill\_plot\_drf
- chill\_plot\_drf\_all
- chill\_plot\_drf\_multi



# iMindMap Branch Attributes



Open Questions  
1. TBS  
2. TBS

Tasks

General Predecessors Resources

Name: <Use Case Set>

Duration: 0.03 day(s)

Start Date:  
Wed 21/09/11 12:00:00 AM

End Date:  
Wed 21/09/11 12:50:18 AM

Priority: 0

% Completed: 0

Mark as Milestone

Enable for Project Management

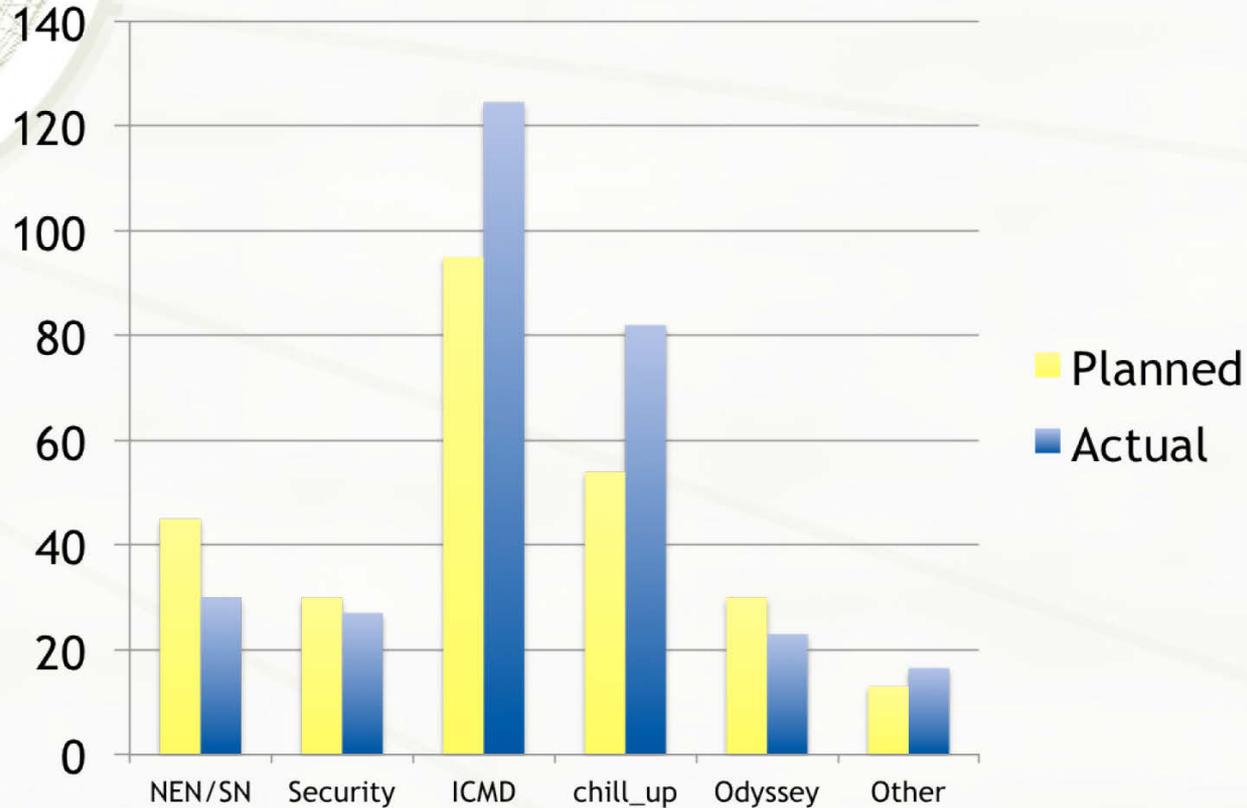


# Results/Lessons Learned

- ✦ An evolving process over the last 3 major releases
  - ✦ Use case definition, more peer review recently added
  - ✦ Use case -> Mind Map -> Development Increment -> Schedule mapping still experimental
- ✦ Mind Mapping very successful at:
  - ✦ Providing a vehicle for bottom-up estimation
  - ✦ Drawing out design details from the entire team
  - ✦ Educating the whole team about the product under development
  - ✦ Preventing needed changes from being “forgotten” in the estimate
  - ✦ Clarifying test cases up front and assisting with test estimation
- ✦ Mind Map estimation not as successful where use cases or peer review are inadequate



# R3 Results



\* Results are "Hours Duration" of tasks



# More Results/Lessons Learned

- ✦ Major schedule milestones met for three releases
  - ✦ Minor variations in increment build schedules
  - ✦ Multiple re-plans occurred for various reasons
  - ✦ Great difficulty tracking actual versus planned in our environment; still working on this
- ✦ Had hoped to largely automate generation of TIPs, Implementation Plans, Schedules from Mind Maps
  - ✦ ThinkBuzan's export and scheduling implementation has problems
  - ✦ Bugs and issues in the product have prevented this
- ✦ Mapping of Mind Maps to schedule items could be confusing: Changes made in the R4 planning to address this