SOFTWARE ENGINEERING AT MOTOROLA SOLUTIONS

Jeff Yakey

Educational background

- 1993 – Graduated from Belvidere High School
- 1997 – B.A. from Wartburg College, Waverly IA
  - Double major in Computer Science and Mathematics
- 1999 – M.S. from Iowa State University in Computer Science
  - Specialized in Algorithmic Robotics

Career at Motorola

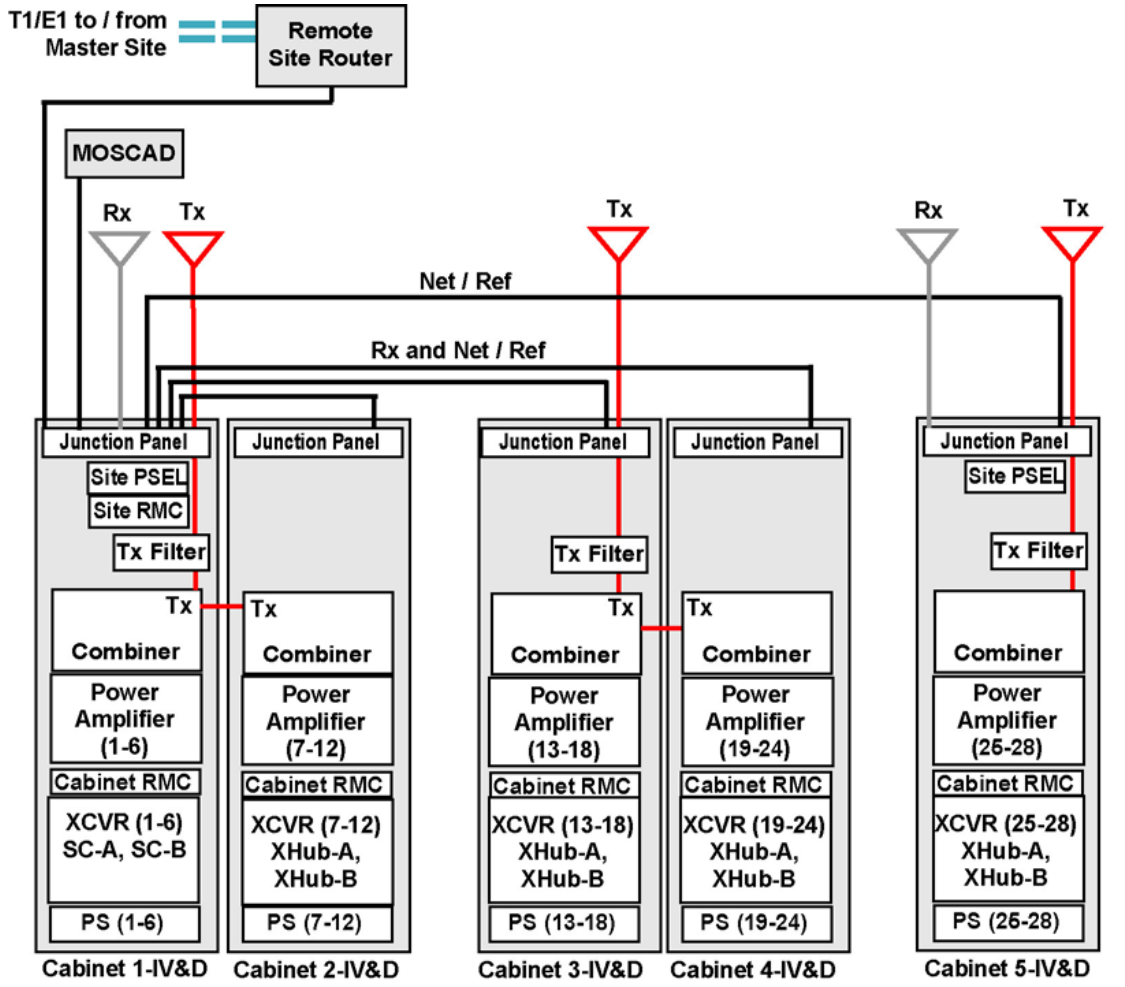
- 1999 – 2002: Japan VLR Group
  - Worked on the MSC/VLR subsystem of the EMX-V CDMA Cellular Switch
  - Embedded real-time software developed in C/C++
  - PPC processor running LynxOS RTOS
- 2002 – 2003: MMSC project
  - Initial development of the Delivery Manager subsystem of the Multimedia Messaging Service Center
  - Developed in C++ on the HP/Tandem NonStop high availability platform (Guardian OS)

Career at Motorola

- 2003 – Present:Repeater Software
  - Product Architect on the G-Series Product Platform, using Enea OSE RTOS
  - High Performance Data Base Radio
    - Public safety TDMA data base radio operating in 25 kHz channels
  - X2 TDMA
    - TDMA voice in 6.25 kHz equivalent channels (2 slots in 12.5 kHz)
  - High Speed Data Base Radio
    - Update to HPD BR to operate in 50 kHz channels
  - Conventional Base Radio
    - Adding analog voice and multi-frequency features

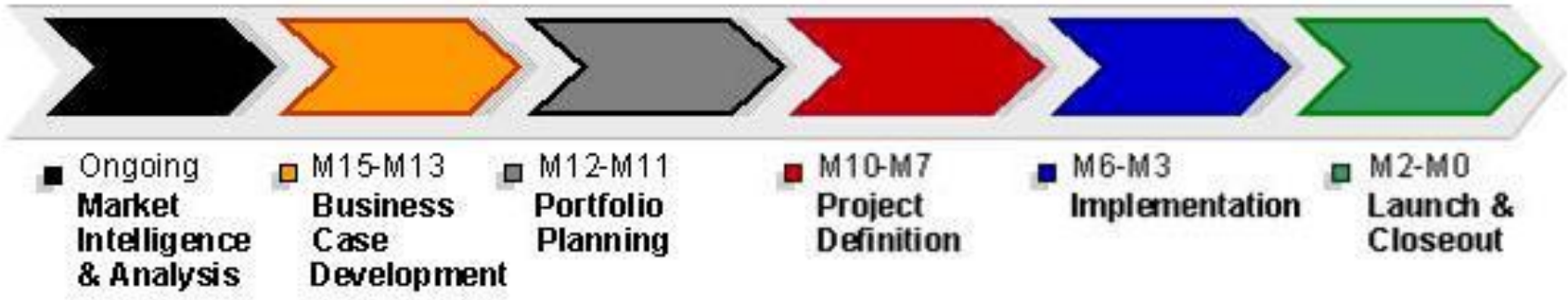
GTR 8000 Expandable Site Subsystem

ASTRO Radio Site Architecture



Product/Project Development Process

Motorola M-Gates



- Corporate wide process for market and product line planning with system and product development teams
- Comprised of 16 gates grouped into 5 phases

M13 - Estimation

- Feature requests are generated by business teams, based on customer requests or market analysis
- Technical experts generate a ROM (Rough Order of Magnitude) estimate for FRs
  - Features are estimated in staff-months
- Business teams prioritize features based on customer need and cost
- Features are selected for a system release and locked down

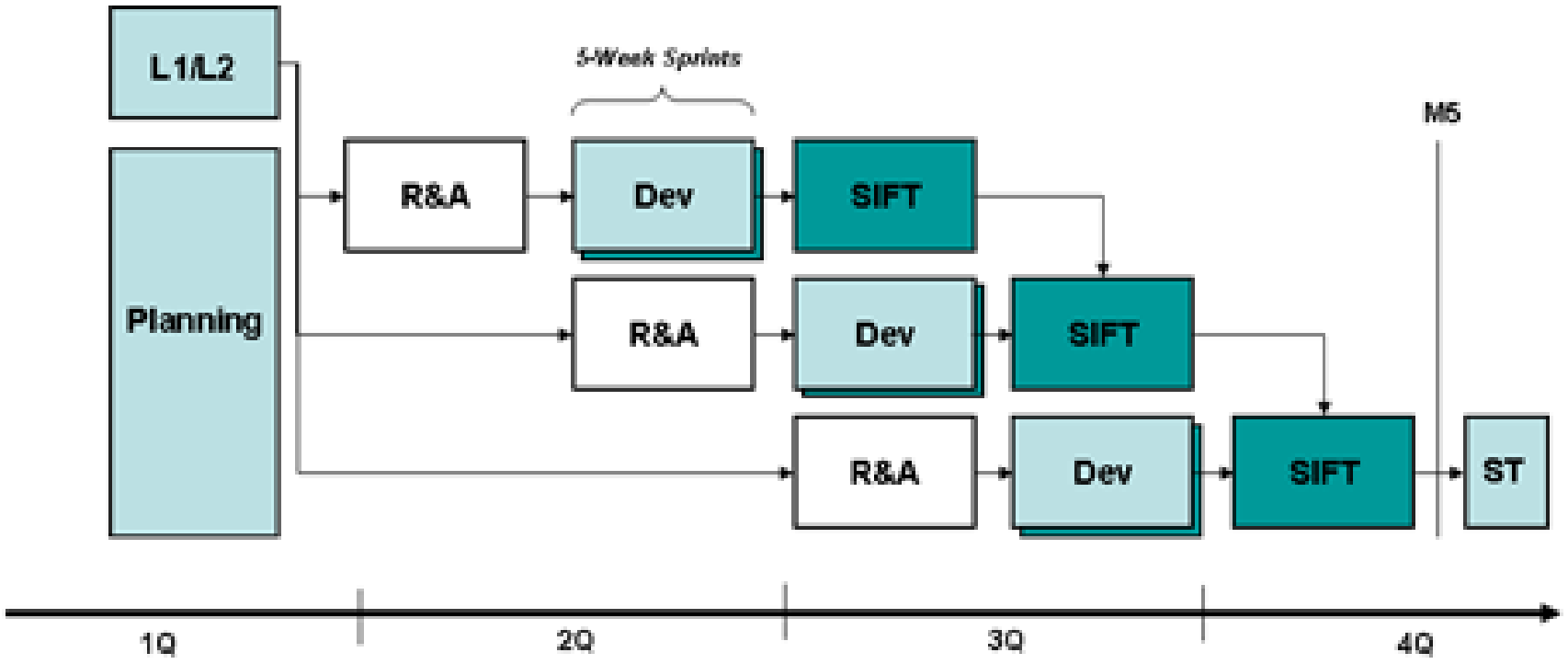
M11 - Scope Lockdown

- System release feature estimates are refined based on further input from system design team
  - High level system design approaches are analyzed
  - Tradeoff is usually between customer needs (extra features) and cost
- Predicted project metrics are generated based on M11 estimates :
  - Project staffing profiles
  - Defect arrival/resolution profiles
  - Test execution profiles

M7 - System Design Complete

- System design team creates L2 requirements, allocated to subsystems
- Feature/product architects participate in L2 design and requirements reviews
- Large features (>500 SM) may follow System Level Agile Process (SLAP)
  - System requirements are developed iteratively
  - Subsystem development teams implement L2 requirements
  - Allows for earlier testing of system integration

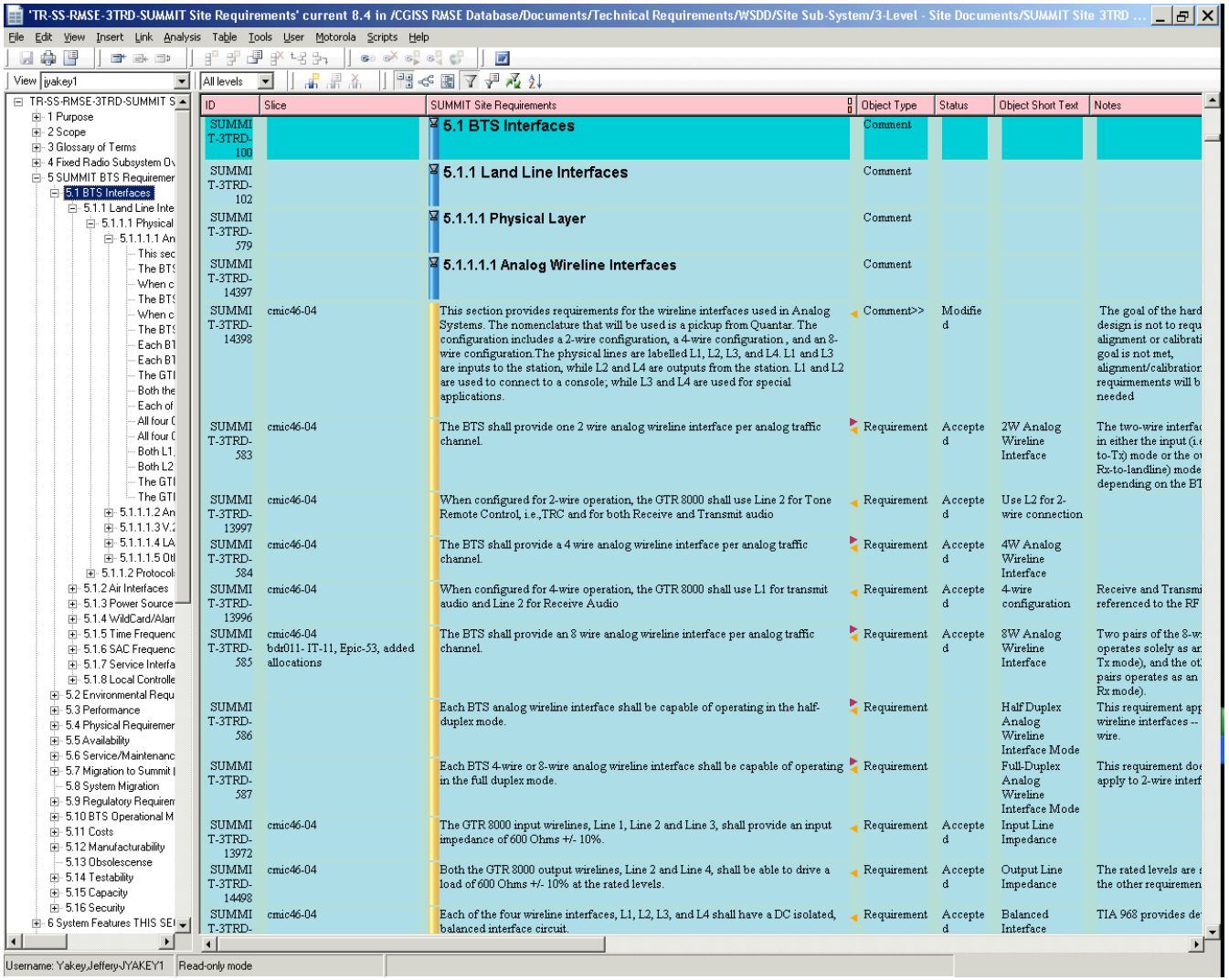
System Level Agile Practice (SLAP)



Requirements creation

- Product level (L3) requirements are captured in Telelogic (now IBM Rational) DOORS requirements management tool
  - Requirements database allows embedding of objects (images, tables, etc.)
  - Provides linkage and traceability mechanisms for upstream and downstream requirements
- Inter-product protocols are capture in Interface Control Documents (ICDs)

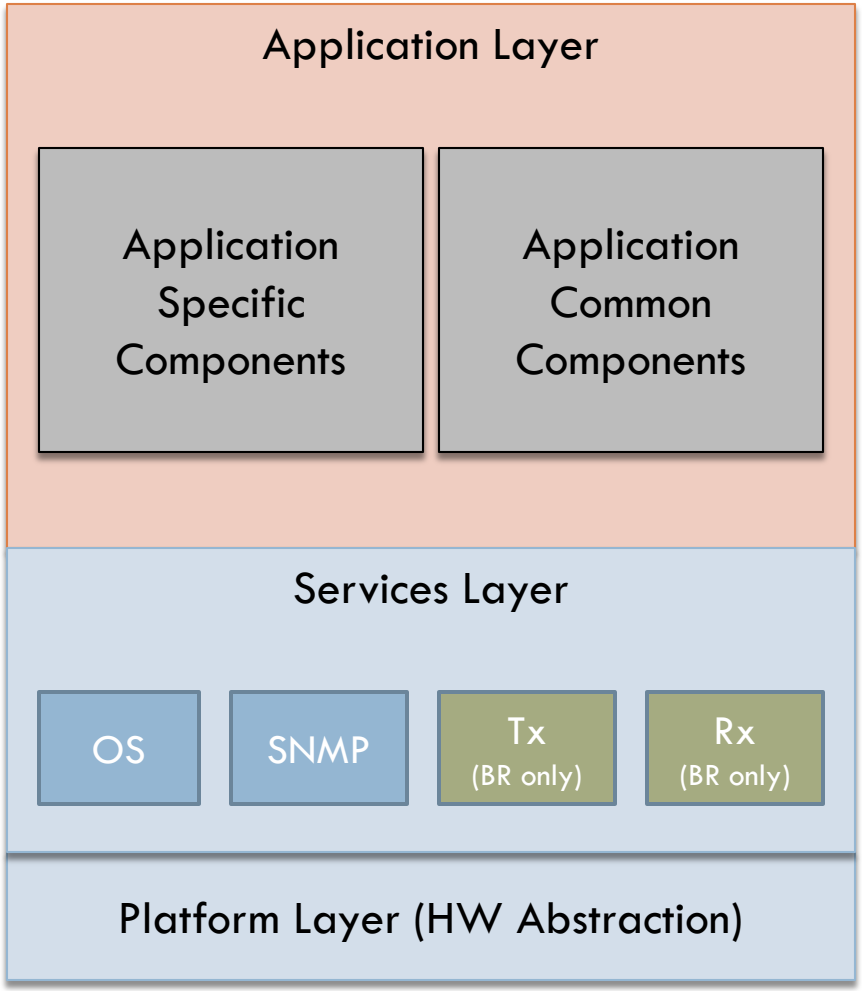
DOORS


'TR-SS-RMSE-3TRD-SUMMIT Site Requirements' current 8.4 in /CGISS RMSE Database/Documents/Technical Requirements/WSDD/Site Sub-System/3-Level - Site Documents/SUMMIT Site 3TRD ...
File Edit View Insert Link Analysis Table Tools User Motorola Scripts Help
View jyakey1
All levels
TR-SS-RMSE-3TRD-SUMMIT S
1 Purpose
2 Scope
3 Glossary of Terms
4 Fixed Radio Subsystem Ov
5 SUMMIT BTS Requiremer
5.1 BTS Interfaces
5.1.1 Land Line Inte
5.1.1.1 Physical
5.1.1.1.1 An
5.1.1.2 An
5.1.1.3 V.
5.1.1.4 LA
5.1.1.5 Ot
5.1.1.2 Protocol
5.1.2 Air Interfaces
5.1.3 Power Source
5.1.4 WildCard/Alarr
5.1.5 Time Frequenc
5.1.6 SAC Frequenc
5.1.7 Service Interfa
5.1.8 Local Controlle
5.2 Environmental Requ
5.3 Performance
5.4 Physical Requiremer
5.5 Availability
5.6 Service/Maintenanc
5.7 Migration to Summit
5.8 System Migration
5.9 Regulatory Requiren
5.10 BTS Operational M
5.11 Costs
5.12 Manufacturability
5.13 Obsolescense
5.14 Testability
5.15 Capacity
5.16 Security
6 System Features THIS SE
ID
Slice
SUMMIT Site Requirements
Object Type
Status
Object Short Text
Notes
SUMMIT-3TRD-100
5.1 BTS Interfaces
Comment
SUMMIT-3TRD-102
5.1.1 Land Line Interfaces
Comment
SUMMIT-3TRD-579
5.1.1.1 Physical Layer
Comment
SUMMIT-3TRD-14397
5.1.1.1.1 Analog Wireline Interfaces
Comment
SUMMIT-3TRD-14398
cmic46-04
This section provides requirements for the wireline interfaces used in Analog Systems. The nomenclature that will be used is a pickup from Quantar. The configuration includes a 2-wire configuration, a 4-wire configuration , and an 8-wire configuration.The physical lines are labelled L1, L2, L3, and L4. L1 and L3 are inputs to the station, while L2 and L4 are outputs from the station. L1 and L2 are used to connect to a console; while L3 and L4 are used for special applications.
Comment>>
Modified
The goal of the hard design is not to requ alignment or calibrati goal is not met, alignment/calibration requirmements will b needed
SUMMIT-3TRD-583
cmic46-04
The BTS shall provide one 2 wire analog wireline interface per analog traffic channel.
Requirement
Accepted
2W Analog Wireline Interface
The two-wire interfac in either the input (i.e to-Tx) mode or the ou Rx-to-landline) mode depending on the BT
SUMMIT-3TRD-13997
cmic46-04
When configured for 2-wire operation, the GTR 8000 shall use Line 2 for Tone Remote Control, i.e.,TRC and for both Receive and Transmit audio
Requirement
Accepted
Use L2 for 2-wire connection
SUMMIT-3TRD-584
cmic46-04
The BTS shall provide a 4 wire analog wireline interface per analog traffic channel.
Requirement
Accepted
4W Analog Wireline Interface
SUMMIT-3TRD-13996
cmic46-04
When configured for 4-wire operation, the GTR 8000 shall use L1 for transmit audio and Line 2 for Receive Audio
Requirement
Accepted
4-wire configuration
Receive and Transmi referenced to the RF
SUMMIT-3TRD-585
cmic46-04 bdr011- IT-11, Epic-53, added allocations
The BTS shall provide an 8 wire analog wireline interface per analog traffic channel.
Requirement
Accepted
8W Analog Wireline Interface
Two pairs of the 8-w operates solely as an Tx mode), and the ot pairs operates as an Rx mode).
SUMMIT-3TRD-586
Each BTS analog wireline interface shall be capable of operating in the half-duplex mode.
Requirement
Half Duplex Analog Wireline Interface Mode
This requirement app wireline interfaces -- wire.
SUMMIT-3TRD-587
Each BTS 4-wire or 8-wire analog wireline interface shall be capable of operating in the full duplex mode.
Requirement
Full-Duplex Analog Wireline Interface Mode
This requirement doe apply to 2-wire interf
SUMMIT-3TRD-13972
cmic46-04
The GTR 8000 input wirelines, Line 1, Line 2 and Line 3, shall provide an input impedance of 600 Ohms +/- 10%.
Requirement
Accepted
Input Line Impedance
SUMMIT-3TRD-14498
cmic46-04
Both the GTR 8000 output wirelines, Line 2 and Line 4, shall be able to drive a load of 600 Ohms +/- 10% at the rated levels.
Requirement
Accepted
Output Line Impedance
The rated levels are the other requiremen
SUMMIT-3TRD-
cmic46-04
Each of the four wireline interfaces, L1, L2, L3, and L4 shall have a DC isolated, balanced interface circuit.
Requirement
Accepted
Balanced Interface
TIA 968 provides de
Username: Yakey,Jeffery-JYAKEY1
Read-only mode

Architecture

- Architecture decomposes feature into major software components
  - Focus is primarily on interfaces and interactions between components
  - L4 Requirements are written to formalize these interactions, and are traced to L3 requirements
  - Goal is to write requirements that can be unit tested
- Architecture and design patterns are used throughout products
  - Layers pattern
  - Active Object and Capsule (UML RT) patterns

GTR Layered Architecture



Repeater Software Agile Process

- Based on Scrum agile process
- Development teams work in 5 week iterations
- Best team size is 4-6 developers
- Iterative Lifecycle
  - Rapid feedback and learning in short cycles
- Collaborative Teaming
  - Teams produce better results than individuals
  - Manage themselves to make better decisions
- Development continuously validates quality

Agile Practices

- Agile Project Management
- Automated Testing
- Continuous Integration
- Customer Proxy
- Daily Stand-Up
- Iterative & Incremental Development (IID)
- Paired Development
- Refactoring
- Retrospectives
- Test-Driven Development (TDD)

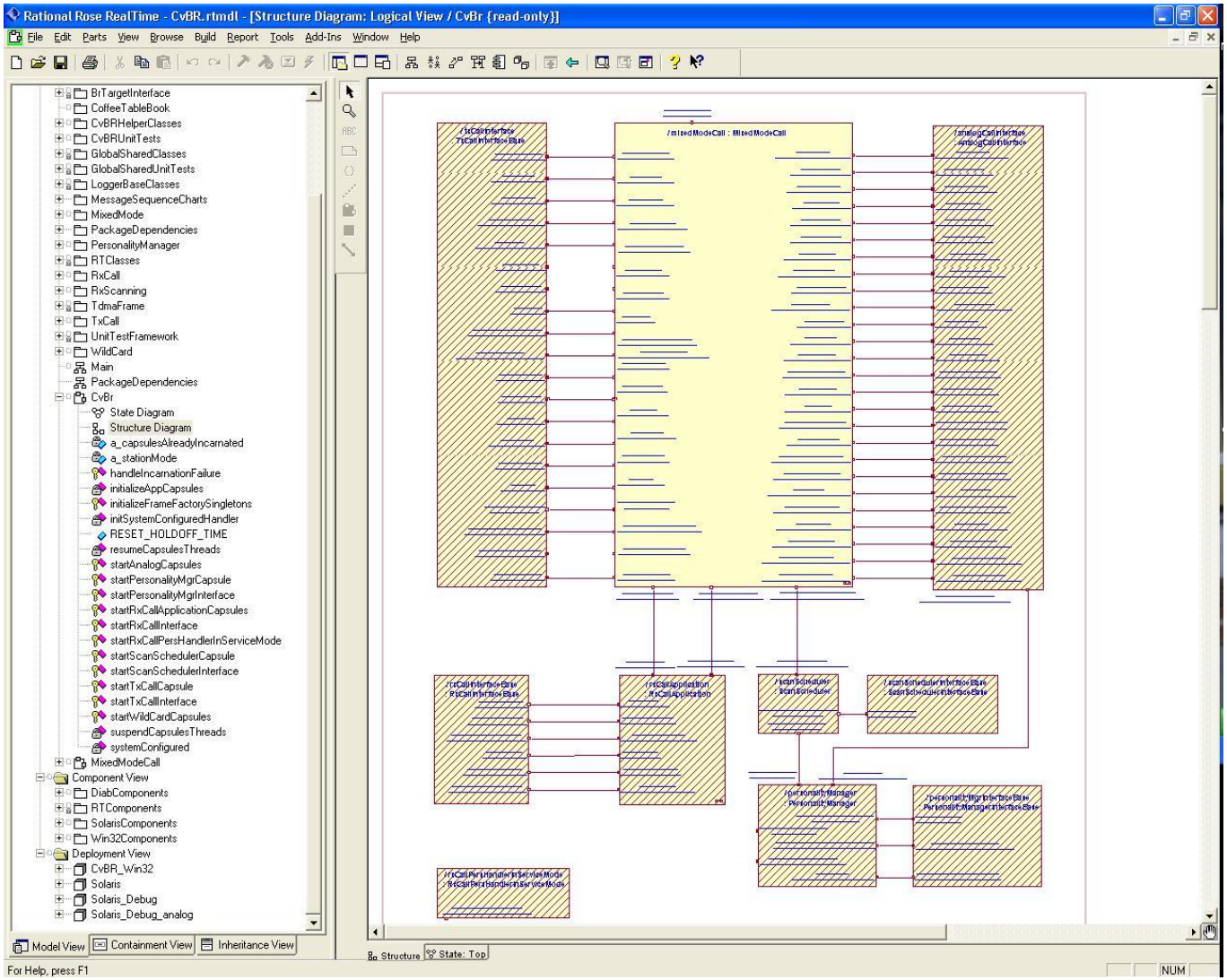
Calendar for Agile Development

5 Week Iteration Calendar							
Sunday	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**Saturday**	
				Team Lead Coordination Meeting	Kickoff		
	Legacy Walkthrough Scenario Scrub	Design Blitz	Planning Game		Informal Design Review		<-- Establish Design
	Planning Game	Start TDD		Design/AR FTR			<-- Lockdown Design
	Planning Game		Complete TDD	Prepare Packets	Code Walkthrough		<-- Implementation
		Test FTR		Code FTR			<-- FTR/Rework
		Postmortem			Kickoff		<-- Pad Week

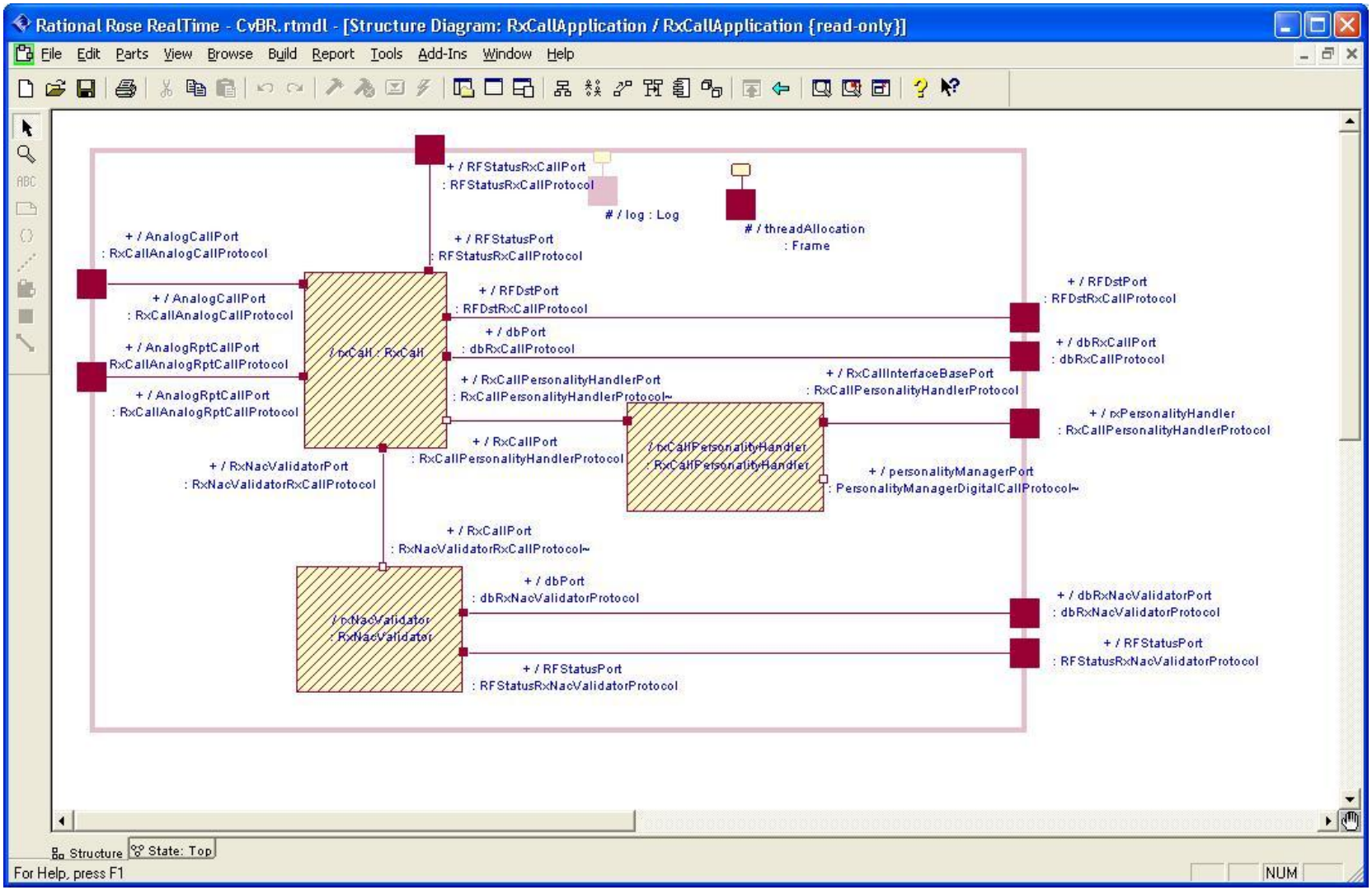
Development tools

- C++/C/PPC Assembly
- Rational Clearcase (Configuration Management)
- Rational Rose Realtime (UML)
  - Auto-code generation
- Cppunit/RoseRT Capsule test framework
  - Automated unit test execution
- Electric Cloud e-make/Commander
  - Automated/parallel build utilities
  - Formerly used CruiseControl
- Codewarrior Debugger/PowerTAP JTAG
- Perl/Python (automation scripts, data processing, testing)
- Wikis

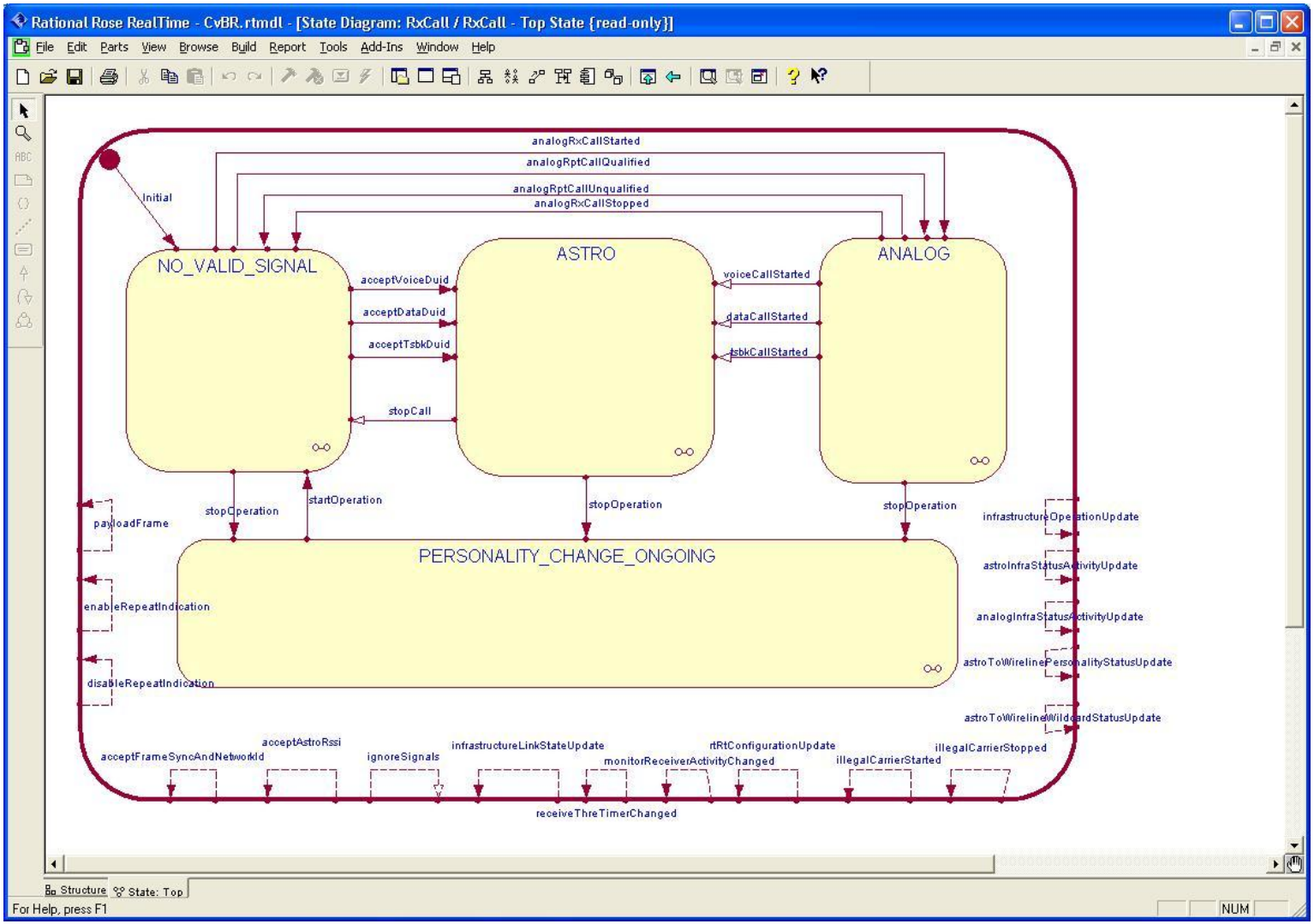
Rational RoseRT – Structure Diagrams



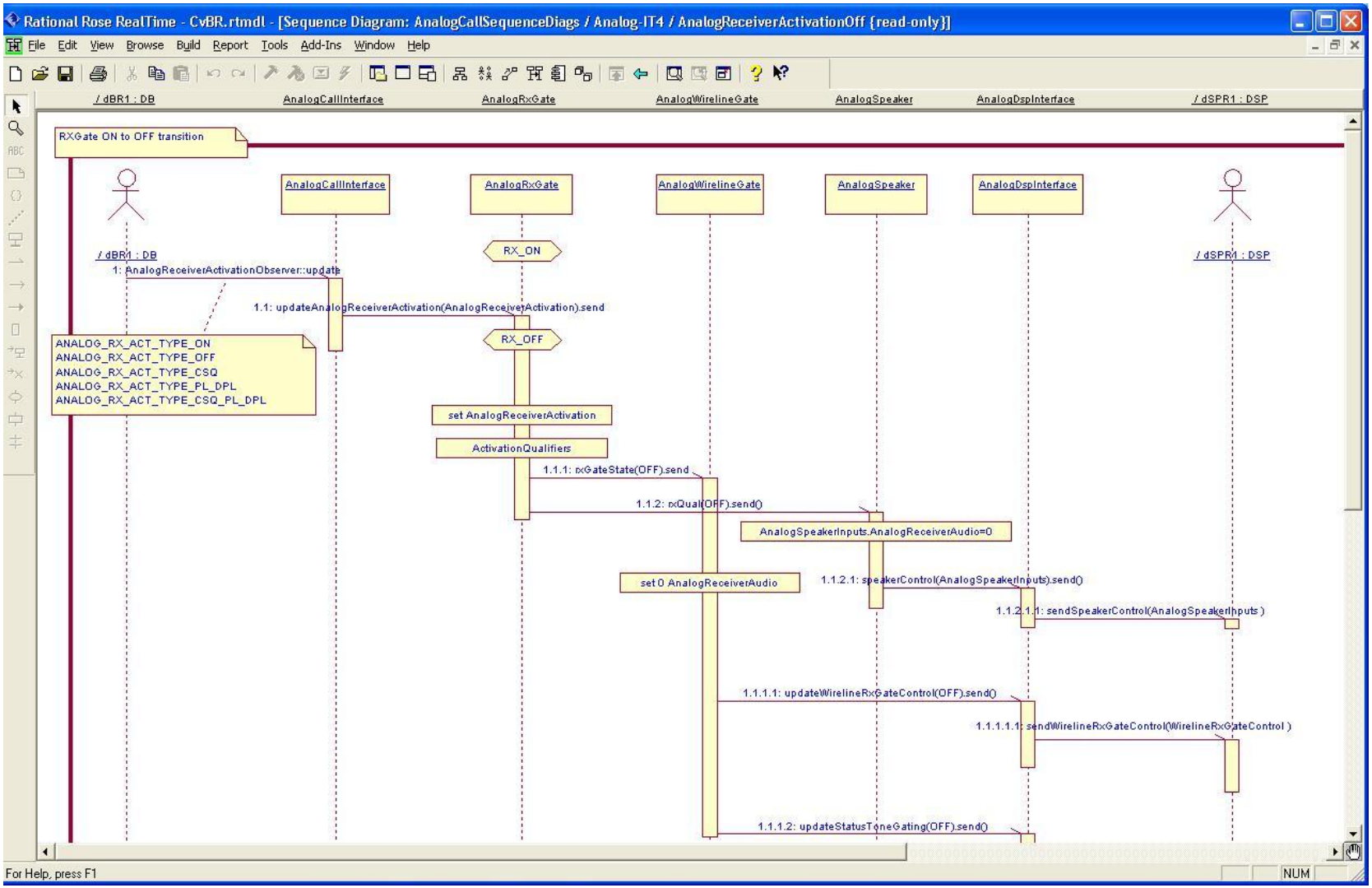
Rational RoseRT – Capsules



Rational RoseRT – State Machines



Rational RoseRT – Message Sequence Charts



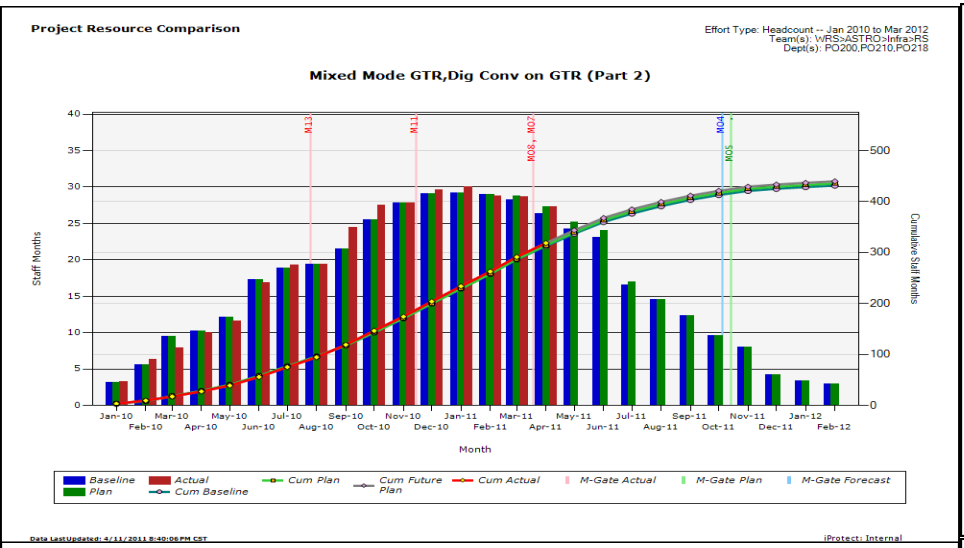
Formal Technical Review (FTR)

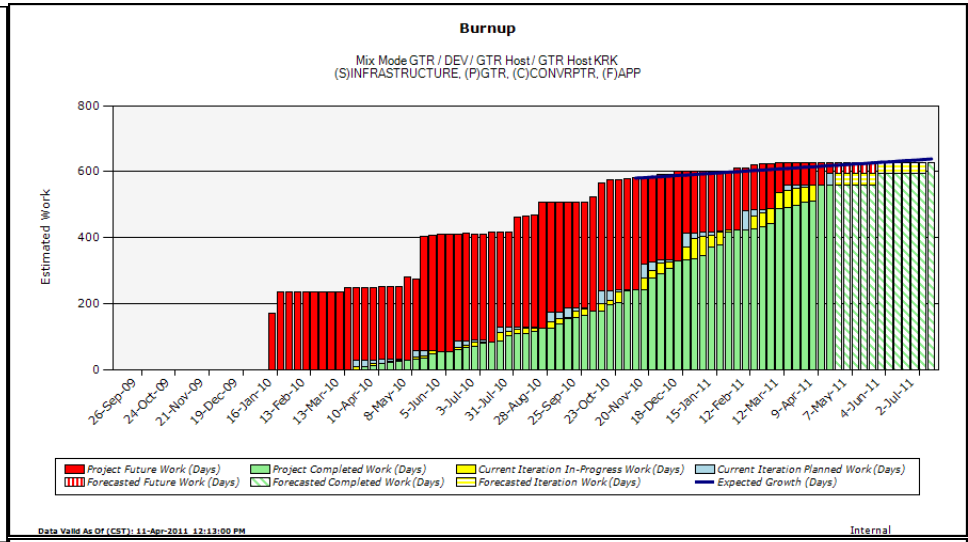
- Peer review process
- All deliverables (requirements, ICDs, design, code) are required to be reviewed
- Improved quality reduces cost of fixing escaped defects

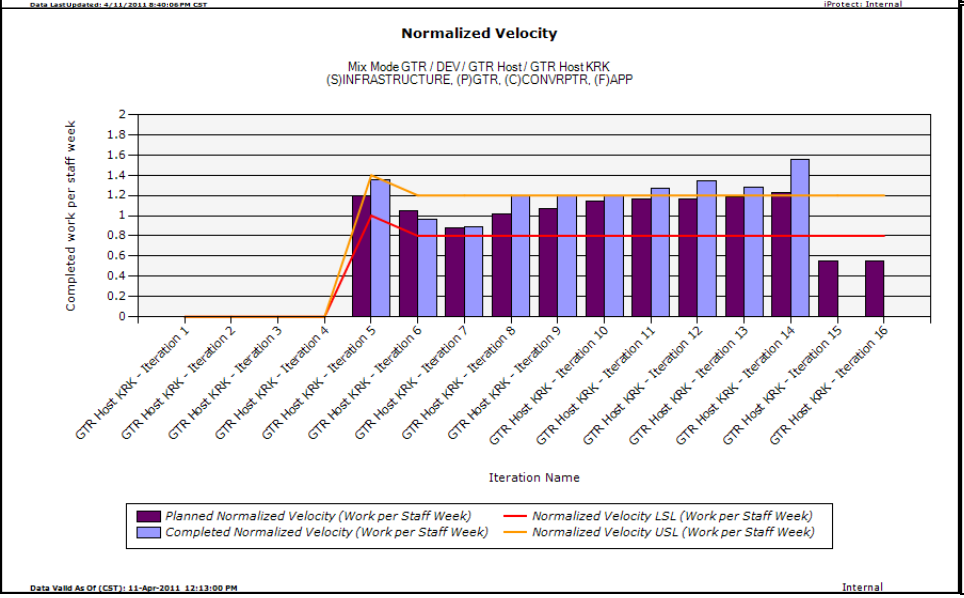
Project Management Tools

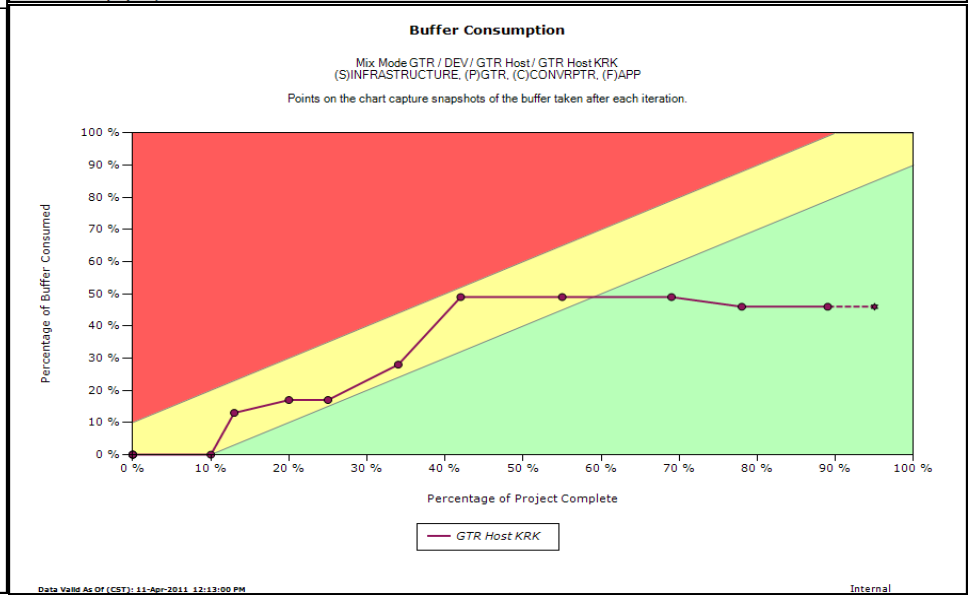
- Version One (Agile Project Management Suite)
- ClearQuest (Defect Management)
- Metrics data collected from tools

Sample Agile Project Metrics

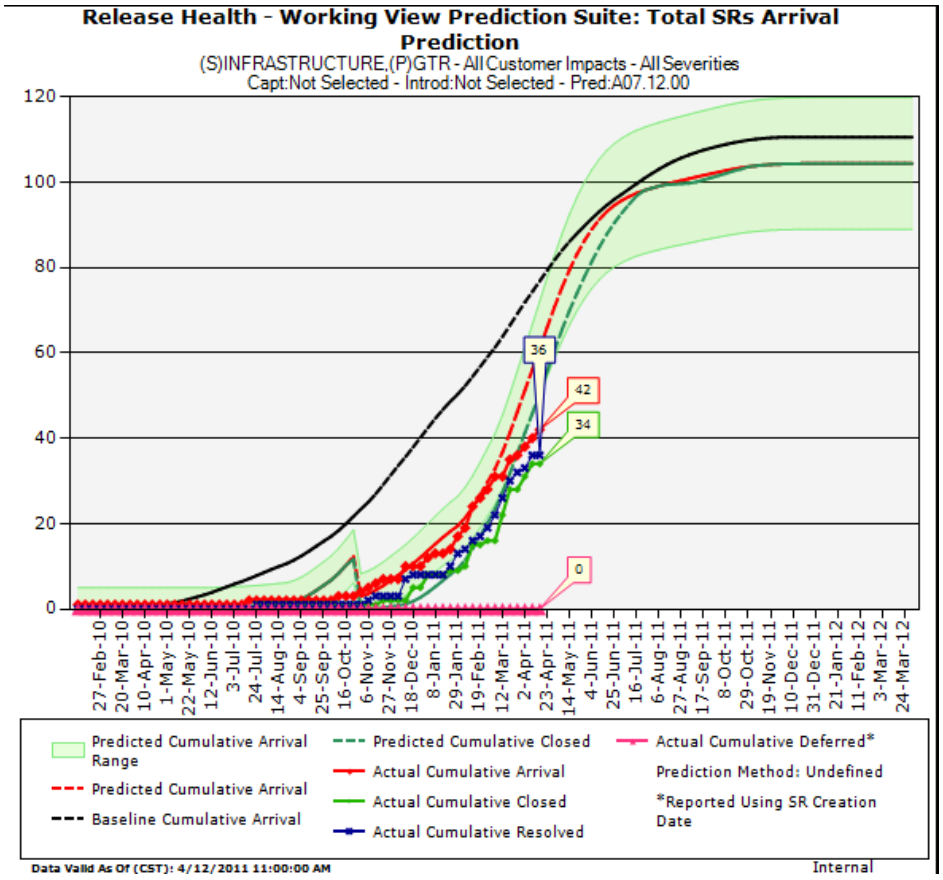


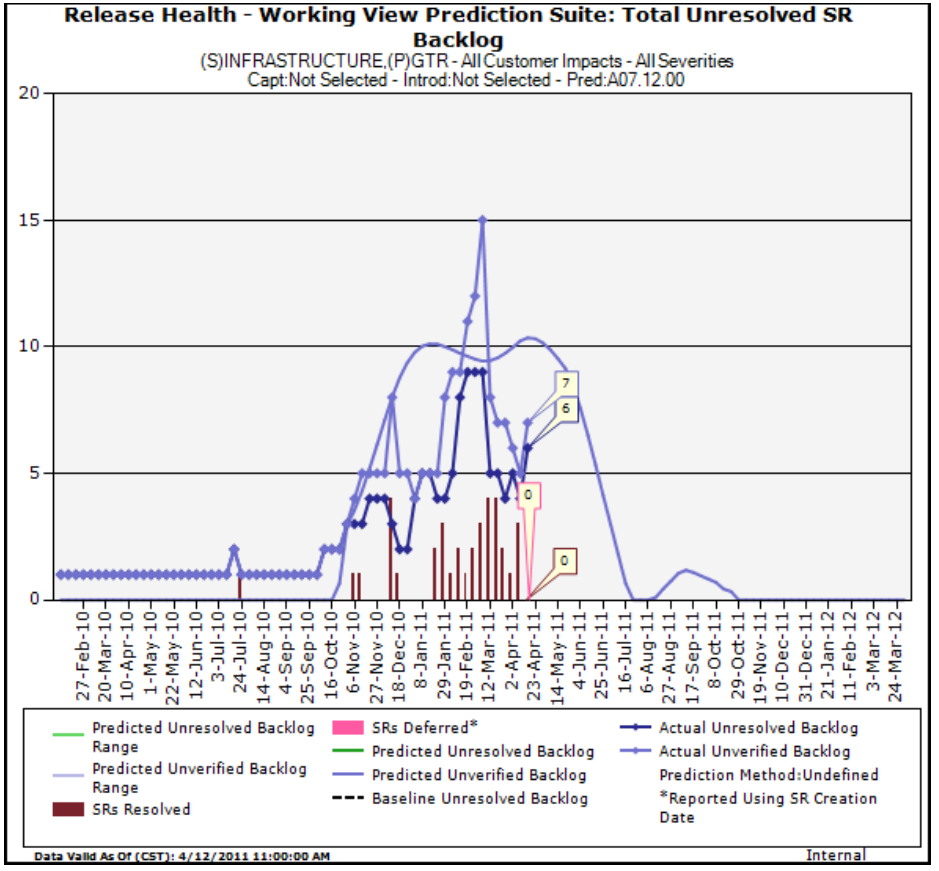






Sample Defect Prediction/Backlog

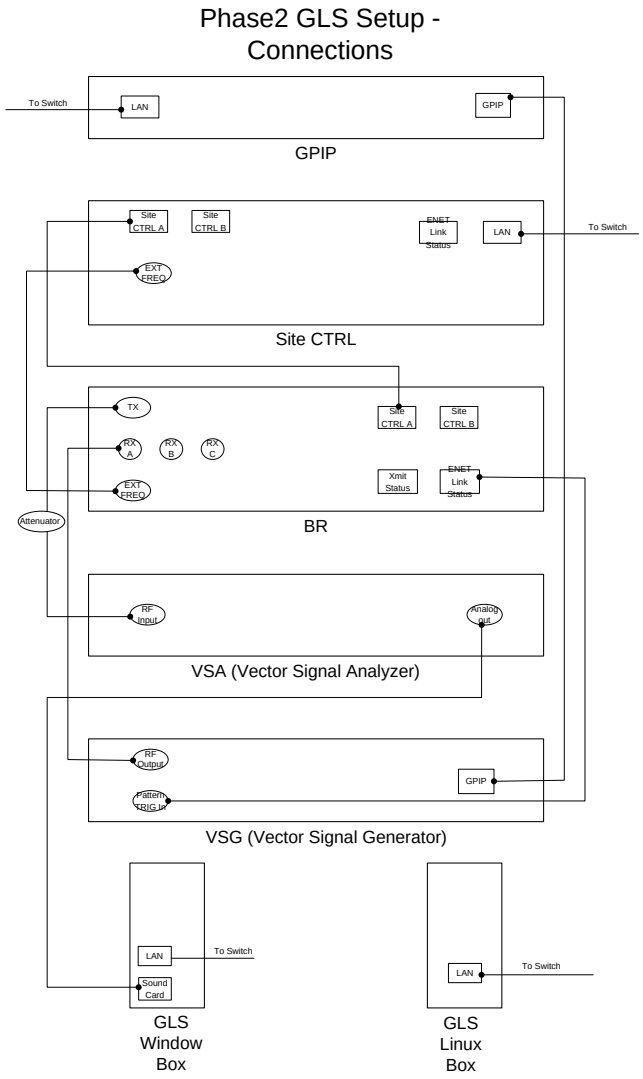




GLS/Box Test

- Generic Link Simulator
  - Motorola developed automated product test framework
  - Primarily intended for network protocol testing (message sending/receiving)
  - Extended to include RF capabilities
    - Vector Signal Analyzer (VSA)
    - Vector Signal Generator (VSG)
    - OTA Protocol Parsers/Formatters

GLS/Box Test Phase



M5 - Product Development Complete

- M5 Goals
  - All feature development is complete
  - All box tests have been executed
  - 90% of box tests must pass
- System test begins
  - Test product and feature interactions
- Primary activities
  - Support system test team
  - Defect Repair

M3/M2 - Product Deployment

- All system testing is complete
  - 100% pass rate is the goal
- Software is turned over to factory
  - Installed on newly shipped systems
  - CDs available for customer upgrades
- Final project metrics collected and archived

Software Engineering Skills

Beneficial Coursework

- Computer languages
  - Exposure to multiple programming paradigms beneficial
  - Use appropriate language to solve problems
  - More OOP/OOD/Design Patterns would have been useful
- Operating Systems
  - RTOS concepts very useful
- Networking
  - Mobile computing is ubiquitous now

Beneficial Coursework

- Statistics
  - Used in performance modeling and profiling
  - Design For Six Sigma (DFSS)
- Communication and leadership skills
  - Essential to be able to collaborate with colleagues
  - Global collaboration is especially challenging
  - Our biggest problems are usually due to a breakdown in communication

Class work is only the beginning

- Continuous learning is essential
- New languages and frameworks keep emerging
  - Skills become obsolete very quickly
- Computing challenges have changed considerably in 12 years
  - Secure programming, Information Assurance, Privacy concerns
  - Multicore and concurrent programming

Questions