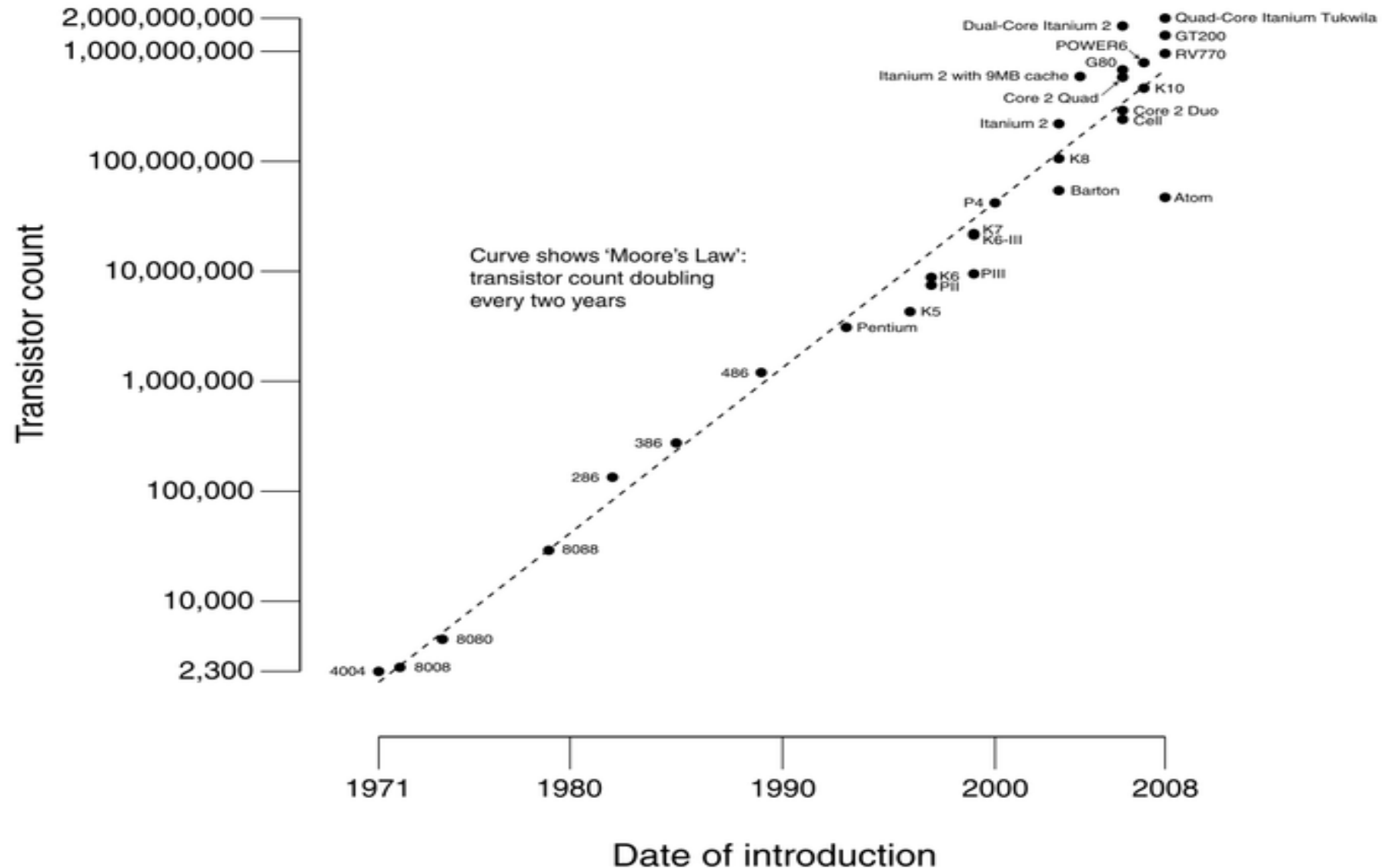


Agile Methods for Verification
using
IVerifySpec

CPU Transistor Counts 1971-2008 & Moore's Law





Semiconductor Industry Challenges

- *80% of designs go over deadline by an average 44%**
 - **70% effort devoted to verification**
 - *80% of product cost is head count related**
-
- Make people more effective
 - Make processes efficient
 - Get appropriate tools

*Source: NXP/Spirit 2008

DAC10 Booth #359

Agnisys

www.agnisys.com

Trends in the industry

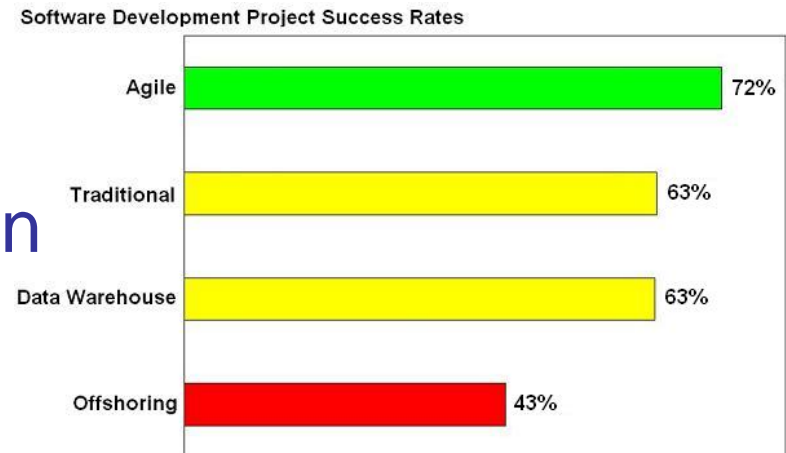
- Higher density
 - Integration of higher density IP in an SoC
- Greater reuse of IP
 - External, 3rd party
 - Internal, older projects, other teams
- Distributed teams
 - Multiple local sites
 - Offshore sites

Management Vs. Verification team

- What Management wants
 - What is the verification status
 - When verification team will be done
 - What is the Quality of Result
- What Verification Team wants
 - Be let alone to do their job!
 - Not do un-productive meetings/status hunts

Why be Agile

- Agile projects are characterized by higher success rates as evidenced by
 - higher productivity
 - better quality
 - lower costs
 - Overall higher satisfaction



Copyright 2007 Scott W. Ambler
Source: DDJ's 2007 Project Success Survey www.amblysoft.com/surveys

Being agile

Main Entry: ag·ile

Pronunciation: \ 'a-jəl, - ,jī(-ə)\

Function: *adjective*

1 : marked by ready ability to move with quick easy grace <an agile dancer>

2 : **having a quick, resourceful and adaptable character** <an agile mind>

— ag·ile·ly \-jə(l)-lē, - ,jī(l)-lē\ *adverb*

Source : www.merriam-webster.com

DAC10 Booth #359

Agnisys

www.agnisys.com

agile Vs. Agile

- **Agile software development** refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.*
- “Agile” is one of the ways to be agile

*Source: Wikipedia

DAC10 Booth #359

Agnisys

www.agnisys.com

Manifesto for Agile Software Development*

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions	over	processes and tools
Working software	over	comprehensive documentation
Customer collaboration	over	contract negotiation
Responding to change	over	following a plan

That is, while there is value in the items on the right, we value the items on the left more.

*Source: agilemanifesto.org

DAC10 Booth #359

Agnisys

www.agnisys.com

Manifesto for Agile Verification

Agile Verification teams value

Individuals and interactions over processes and tools,
... except where the tools help individuals and interactions

Controlled regression environment over comprehensive
documentation

Design team collaboration over contract negotiation

Responding to change over following a rigid
verification plan

... except when the plan can respond to change

Where does change come from?

- Requirement changes
- Design changes
- Changes to the verification strategy
- Updates to verification plan as the project matures
- Resource changes which has an impact on verification priority and what gets verified
- Schedule and milestone changes
- Change of simulation vendor

Verification management challenges

- **Verification planning**
- **Constantly evolving and highly dynamic verification**
- **Too much data, everywhere**
 - **Regressions**
 - **Coverage reports**
 - **Log files**
 - **Bug tracking system**
- **Versions/configurations of design files, software revisions, ...**

Verification management challenges (contd.)

- **Dispersed data sources**
 - *Verification using various methodologies*
 - *Variety of vendors with their specific formats*
 - *Other sources of information*
 - Bug tracking systems
 - Project management system
 - Requirement capture system
- **Distributed design and verification teams**

Possible solutions

- Static documents and spreadsheets
- Using Wiki
- Home-grown custom application
- IVerifySpec

What is IVerifySpec™

- Verification planning and management tool
- Enables quick assessment of verification status
- Gives deep insight into the project progress
- Integrates
 - Design requirements
 - Verification plan
 - Results from the verification environment
 - Bug tracking systems

IVerifySpec Overview

Requirement Document

REQ.1 An interleaver scrambles the byte order of incoming data in order to aid error detection and correction schemes such as Reed Solomon/Viterbi. In the design used for this lesson, the incoming data consists of a sync byte (0xb8, 0x47) followed by 29 bytes of packet data.

REQ.2 How many times the Sync byte(0Xb8,0X47) and data have been generated out by downstream rdy_acpt block

REQ.3 The 203 bytes consist of 187 bytes of data to which a Reed Solomon encoder has previously appended 16 bytes of data.

204 Bytes

Sync byte	187 bytes payload data	16 bytes RS encode
-----------	------------------------	--------------------

Plan Document

REQ.1 An interleaver scrambles the byte order of incoming data in order to aid error detection and correction schemes such as Reed Solomon/Viterbi. In the design used for this lesson, the incoming data consists of a sync byte (0xb8, 0x47) followed by 203 bytes of packet data.

name	Interleaver Scrambler					
description	How many times the Sync byte(0Xb8,0X47) and data have been generated out by downstream rdy_acpt block					
requirements	1					
features	description	link	type	curr.	prev.	bugs
Upstream coverage	Upstream packet coverage	up_covg	CoverGroup	100.00	98.30	
Upstream data	Coverage for data	upcov_data	CoverPoint	100.00	100.00	
Upstream Sync	Coverage for Sync byte(0xb8,0x47)	upcov_sync	CoverPoint	100.00	100.00	
Upstream delay	Delay associated with each byte transferred	up_delay	CoverPoint	95.00	95.00	0

REQ.2 How many times the Sync byte(0Xb8,0X47) and data have been generated out by downstream

Web Interface

IVerifySpec™ : Web Interface
Version: 1.1.3

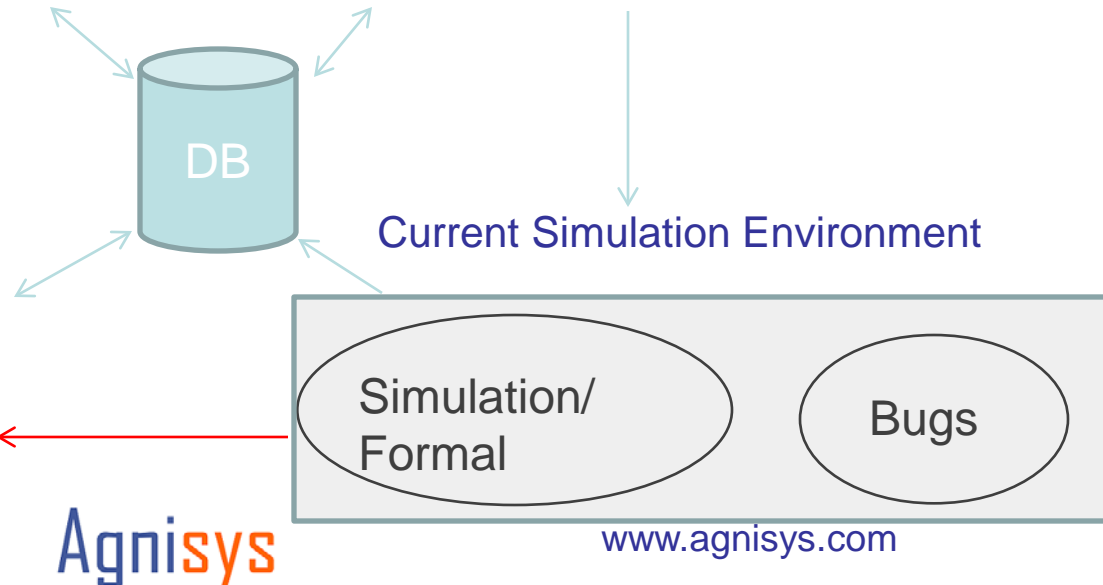
Projects Home Project Details > Graphical Analysis

Document Summary Graphs:

- Current Requirements Summary
- Plan Document Overview [Beta]

Current Requirement Summary
Lastest requirement(s) with Testcase and Bug Count

Requirement ID	Total Count	Testcase Count	Bug Count
1	6	1	0
2	6	3	0
3	6	0	0
4	4	1	0
5	7	1	0



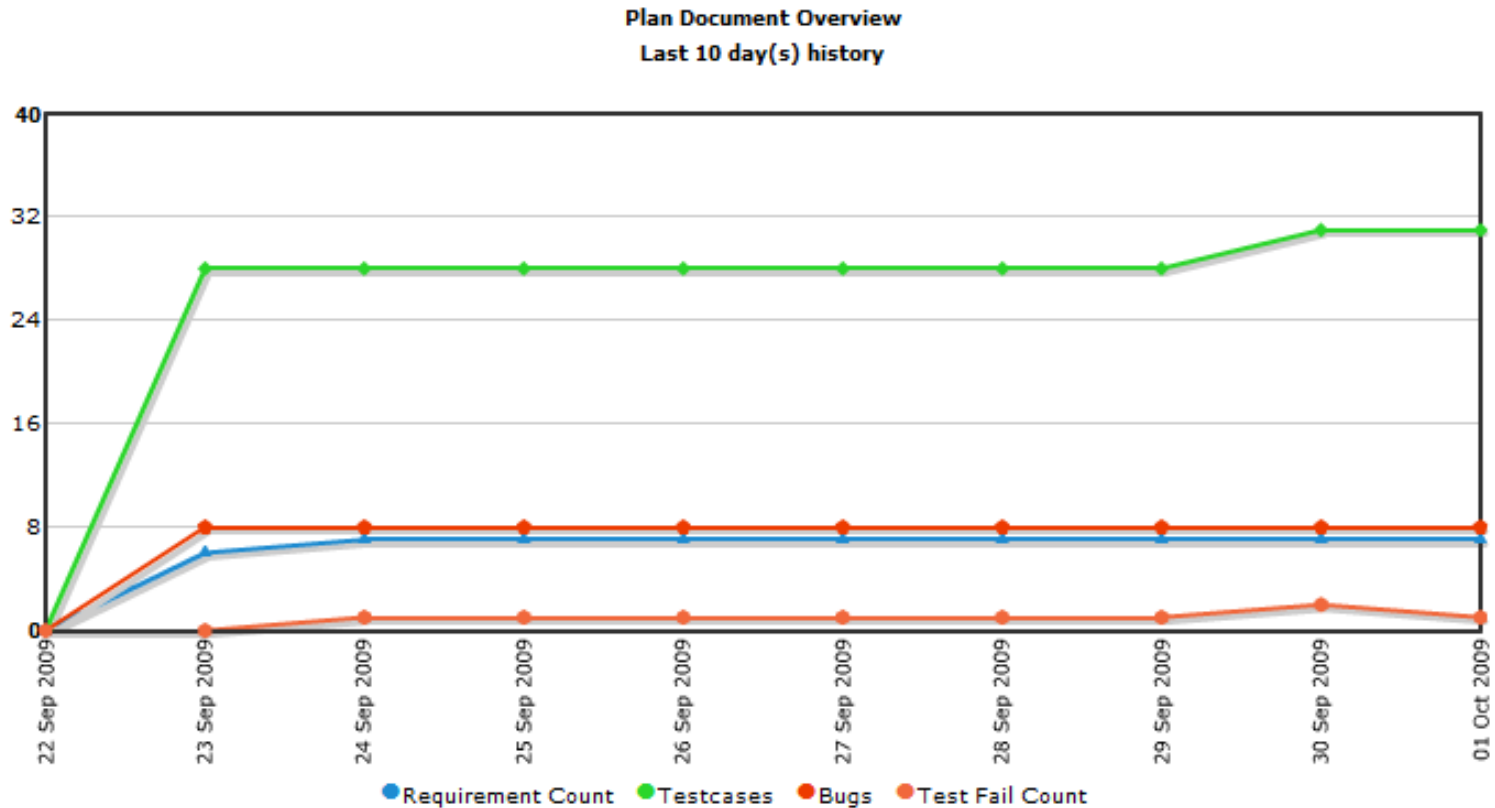
IVerifySpec (contd.)

- Centralized portal for Verification team and management
- Consolidated Information sources
 - Simulation/formal environment
 - Bug tracking system
 - User defined
- Simulation tool vendor or methodology agnostic : Works with all vendor tools!
- Browser based OR Document based Verification plan and requirements
- Collaborate on an intranet and automatically generate MS Word document for plan/requirements
- Deep analysis and charts

IVS Dashboard



Analysis (Historic trends)



Analysis (Heat-map)

Last 10 Day(s) History										
	Sep, 23	Sep, 24	Sep, 25	Sep, 26	Sep, 27	Sep, 28	Sep, 29	Sep, 30	Oct, 01	Oct, (latest) 02
Test2										could not run
add_cp										100
di_data_hold										Pass
di_handshake										Pass
dn_cvg										100
dn_delay										100
dncov_data										100
dncov_sync										100
do_data_hold										Pass
do_handshake										Pass
in_hs										100
in_hsXint_state										100
int_state										100
int_state										100
int_state										100
out_hsXint_state										100
pkt_start_check										Pass
push_mutex_check										Pass
ram_cvg										100
ram_read_check										pass
ram_write_check										pass
s_interleave_sm										100
sm_cvg										70
sm_transitions_cvg										100
test1										pass
test3										pass
up_cvg										100

Verification
Performance
Indicators

No Simulation runs!

More
Green
is
Good

Summary

- Using agile
 - Reduces TTM
 - Increasing the efficiency of the Verification and Design teams
 - Everyone on same page, with clear understanding
 - Better communication
 - Reduce costs
 - Projects done faster
 - Using less simulation licenses
 - Compute resources
 - Improve Quality
 - Verification progress is not a surprise
 - Everyone gets a clear picture including mgmt.



IVerifySpec™

Agile Verification Management

Verification Closure

Verification Dashboard

Agile Friendly

TTM

Goals

Requirements

Documentation

Show-stoppers

Priorities

Customers

Assignment

Historic Trends

MANAGEMENT



Verification Closure

Collaborative Planning

Simulation Vendor Neutral

VERIFICATION TEAM

UVM Sim Results
 5,000 Tests
 VMM OVM
 89% Coverage VDB
 Regressions seed
 Bugs Random
 Log Files Directed
 CVS Mantis
 UCDB VDB
 Formal UCIS
 Bugzilla XML

Are we done yet?

Can we have more ...?

Further information

- <http://agilemanifesto.org/>
- <http://edadesignline.org> for an article on
“Achieving verification closure with resource and time constraints”
- <http://agnisys.com>
- <http://agileSoC.com>

About Agnisys

- Privately held, offices in US and India
- Founded in 2007
- What we do
 - EDA tool development
 - IDesignSpec
 - IVerifySpec
 - Verification of ASICs/FPGAs
 - Video, Audio, Image processing, Codecs, Networking, SONET

IVerifySpec

Agile Verification Management

To learn more ...

Stop by at **Agnisys Booth #359** to sign up for free evaluations