Analytical study of software companies status report regarding software products during 2001 to 2018

Dr. Maneesh VilasDeshpande
Assistant Professor
Maneeshvdeshpande@rediffmail.com

Received: April 21, 2019 Accepted: May 24, 2019

ABSTRACT: *Today's World Is A World Of Computer. Software Is Always Changing, And Its Impact On All Aspects Of Human Life Is Growing. The Increasing Need Of Software Puts Extra Pressure On The Software Companies. And Because Of That Extra Pressure, Less Time, Changing Requirements Of User's It Is Difficult To Create A Software Product On Time, Within Budget And Having Less Number Of Bugs. So This Paper Focuses On Such Type Of Problems Those Are Responsible For The Same. The Overall Analysis Took Place On The Sample Table, Which Is Send To Different Companies And Depends On The Companies Reply Try To Study The Different Reasons And How We Can Cope With The Situation.*

Key Words: *Software Development Life Cycle (SDLC), Software Testing, Bug, Abandon Project, Over Budget And Time Overrun.*

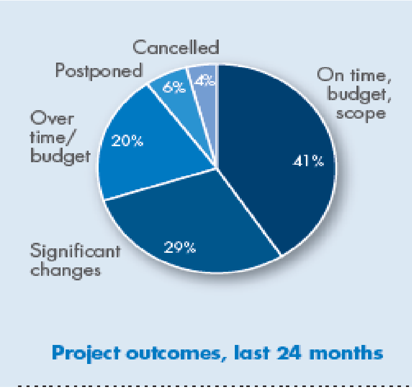
Introduction: Today's World Is A World Of Computers**,** Software Is Always Changing, And Its Impact On All Aspects Of Human Life Is Growing. We Can't Picture A Day Without Cell Phones, Internet Access, Sending E-Mails, And Watching Television, Among Other Things. All Of These Tasks Rely On Software, And Software Bugs Can Be Found Almost Anywhere. Fritz Bauer [FRIT68], In His 1968 Concept Of Software Engineering, Characterized Software Engineering As "The Establishment And Use Of Sound Engineering Principles In Order To Obtain Economically Developed Software That Is Reliable And Works Efficiently On Real Machines". "A Discipline Whose Goal Is The Production Of Quality Software," According To Stephen Schach [SCHA90], Software That Is Delivered On Time, Within Budget, And That Satisfies Its Requirements" Both Definitions Are Widely Used And Accepted. The Attitude Of Software Corporations Is Fairly Apparent. The Main Goal Of Software Firms Is To Provide A Software Product On Schedule, On Budget, And With As Few Flaws As Possible. Software Firms Must Adopt Software Engineering Concepts, New Methods, And Avoid Disagreements Among Team Members In Order To Achieve This. The Increasing Need Of Software Puts Extra Pressure On The Software Companies. And Because Of That Extra Pressure, Less Time, Changing Requirements Of User's It Is Difficult To Create A Software Product On Time, Within Budget And Having Less Number Of Bugs. And Due To The Same Above Problems Many Software Projects Are Not Following The Schedule, Budget And Some Are Abundant, Have To Restart And Cancel As Well. Software Companies Are To Blame For The Problems Because They Rush Goods To Market Without Thoroughly Testing Them. [1]. According To An IBM Research, "31% Of Projects Are Discontinued Before They Are Completed, 53% Overrun Their Cost Projections By An Average Of 189 Percent, And 94 Projects Are Restarted For Every 100 Projects" [IBMG2K]. This Study Is Organized As Follows: Earlier Work (Literature Review), All Important Findings, Conclusion, Acknowledgement, And All-Essential References.

Literature Review: The First Reference Involves The Evolving Role Of Software, Including The Software Failures, And Various Software Engineering Definitions. This Paper Concentrates On The Amount Of Phrases That Can Be Used To Improve The Software Product. [1].The Next Paper Includes The All Important Findings About The Project Outcome, Most Common Cause Of Project Failure And Most Importantly Priorities Assigns When Project Comes Under Pressure. This Paper Also Include The Different Ways How We Can Be Used The Testing In The Organization. Manual As Well As Automation Testing Considered About Organization [2]. October 16, 2006 [3], Mark L. Gillenson, Phd, CCP Proposal For Research At The University Of Memphis The Authors Of "Engaging Testers Earlier In The SDLC" Hold The Following Opinions: We Believe That Software Development Is Fertile Ground For Cross-Functional Teams, With Testers Serving As Vital Members Of Such Teams Throughout The Development Process. Testers Perceiving Themselves As Stakeholders In The Quality Of The Finished Applications As A Result Of Their Efforts Across The SDLC Will Make An Additional Contribution. As A Result, Systems Testing Will Become A More Well-Known And

Respected Speciality Inside Information Systems Businesses. Finding And Correcting These Issues Early (I.E. During The Requirements Or Design Phase) Lowers The Overall Risk And Expense Of The Product [4], And This Study Focuses On The Software Inspection Technique. Next Paper Focuses On Improving ISO 9126 [5] And Monitoring Software Quality During The Life Cycle Of Software Development. The Author Emphasises The Necessity Of Testing Throughout The Software Development Process [6] And Argues That Software Testability Analysis Can Help Estimate The Likelihood That Flaws Are Not Hiding After The Current Version Has Passed All Tests. This Paper Explains How Software Testing Alters Its Nature In Terms Of Working From One Organisation To The Next. To Prevent Problems, It Is Best To Start The Testing Process Early In The Software Development Life Cycle, Or SDLC [7]. Testing Is The Single Most Expensive Component In The Creation Of Advanced, Software-Intensive Military Systems. Significant Cost Savings Can Be Realised If The Concept Of Testing Is Implemented Early In The Development Process. [8]. The Focus Of This Paper Is To Present The First Phase Of Development Of Decision Models That Could Be Used To Determine The Best Uses Of Software Testing Resources. The Ultimate Model Could Be Used To Reduce Overall Costs While Applying Resources Where They Provide The Greatest Value Throughout The Systems Development Process [9].

Findings:

A) Project Outcome: Despite What Has Been Learnt About The Prerequisites For Successful Software Development Outcomes, Project Failures Continue To Be Unacceptable. The Percentage Of Projects Finished On Time, On Budget, And With No Substantial Changes In Scope Declined This Year To 41% From 48% In 2010, While The Percentage Of Projects With Significant Changes Increased From 22% In 2010 To 29% This Year [Adopted From 2]



Our Findings: The Collection Of Data In This Paper Was From 2001 To 2014, So That Able To Study On Large Data. In This Paper We Consider Different Environment So That The Research Is Not Restricted Any More. Outcome Of The Search Is As Shown In The **Table-1** Below With The Concern Diagram.

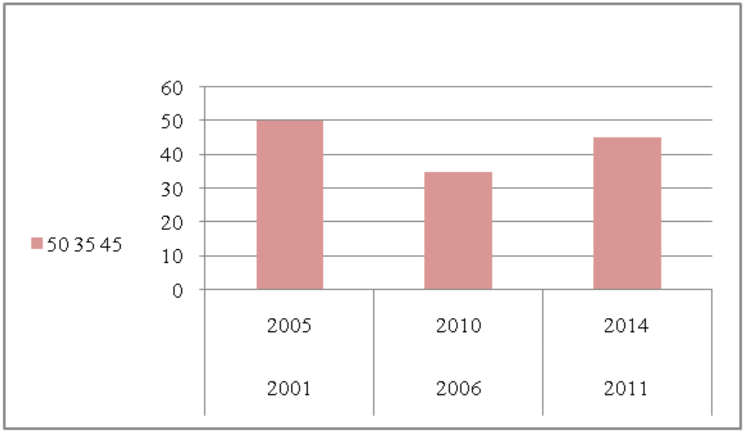
Sr.No	Considered Year		Development Type	**Completed Within Time& With Real Cost**	Testing Ratio		Out Of Total Consider Projects
					After Testing	After UAT	
1	2001	2005	SDLC	50	30%	70%	301
2	2006	2010	SDLC	35	40%	60%	
3	2011	2014	SDLC	45	45%	55%	

Table-1: Project Completed Within Time And With Real Cost

We Consider Different 15 Companies In All Over India. Out Of 11 Companies Are Explore The Data And 04 Companies Not Revealed The Data Because Of The Security Purpose. In This Data Collection We Never Need The Projects Name, Since Sometime It May Concern To The Company But Only Want To Gain Some Statistic About This Major Portion And The Finding The Cause For The Same. The Above Table Shows The Data From 2001 To 2014. For Consideration We Divide Those Years In Three Different Categories Staring From 2001 To 2005, 2006 To 2010, And 2011 To 2014. So Out Of 301 Overall Projects Only 130 Projects Are Completed Within Time And Within Real Cost. Since These Companies Used The Software Development Life Cycle Approach, And In That The Software Development Life Cycle Approach And In That

The Software Development Life Cycle Approach And In That The Tester Term Is Used At The End Of The Development So That The Percentage Of Bug Is Less And After Actual Implementation The Ratio Is Looking Somewhat Interesting. If Some New Approaches If Used Then This Ratio Could Be Reverse And For Same Thing To Happened Always Need To Use Software Testers As Early As Possible. So That The Bug Ratio Could Be Minimized And We Have The Quality Software, And Due To That The Re Cost Value Is Also Seems To Be Less.

The Concern **Graph-1** Is Also Shown Below, Where We Considered Only Two Quantities. On The X-Axis Considered The Years And On The Y-Axis Considered The Completed Projects.



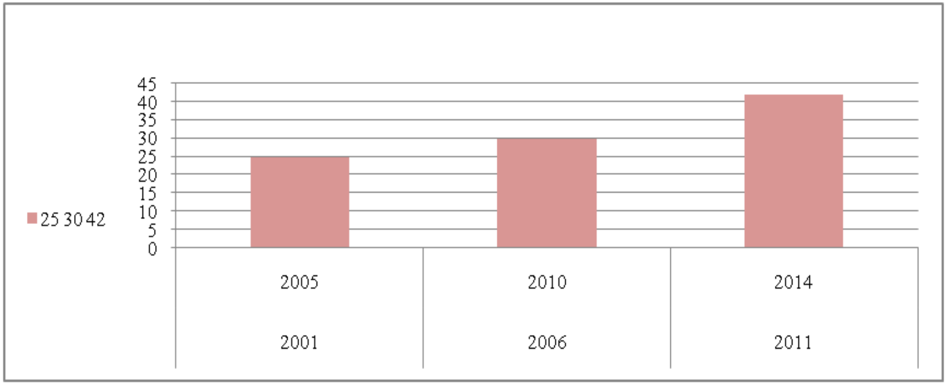
Graph-1 : Total Completed Projects Out Of 301

After Total Completed Projects We Shift Our Focus On How Many Projects Are Not Completed During These Years. And The Findings We Get Are Summarized The **Table-2**as Follows. Staring From 2001 To 2014out Of 301 Total Projects 97 Projects Are Not Completed Within Given Time I.E. Not Able To Meet The Given Deadline. Because Of The Project Overrun The Cost Related With Those Project Are Also Increase And Those Projects Shifted To The Next Level Over Run Projects With Re Cost. The Percentage Of Re Cost Value Increases Since The User Requirements Are Not Changing Continuously In Early Considered Years. But By Years Goes On The Users Are More Advanced And Wanted To Use New Technology With As Always Changing Requirements.

Sr. No	Considered Year		Development Type	**Not Completed Within Time**	Over Run Projects With Re Cost	Out Of Total Consider Projects
1	2001	2005	SDLC	25	15%	301
2	2006	2010	SDLC	30	37%	
3	2011	2014	SDLC	42	48%	

Table-2: Overrun Projects With Re-Cost.

The Concern **Graph-2** Indicates The Same. On The X-Axis Considered The Years And On The Y-Axis Considered The Completed Projects Those Are Overrun.



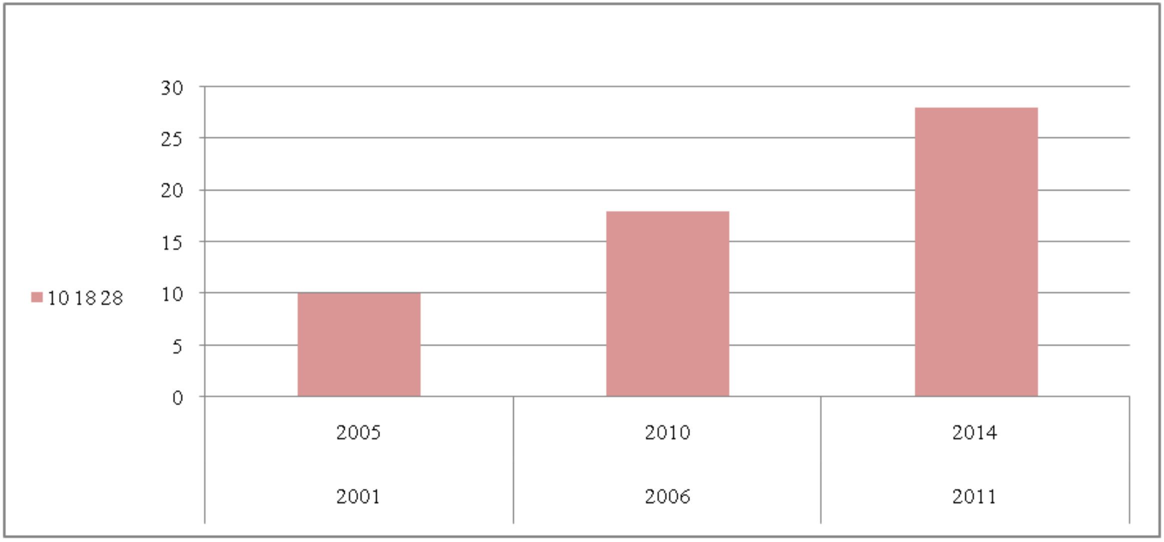
Graph-2 : Not Completed Projects Out Of 301

Table-3 Shows That The Projects Are Abandon During The Year 2001 To 2014 Out Of 301 Overall Projects. Where All The Companies Still Relay On Software Development Life Cycle

Sr. No	Considered Year		Development Type	**Abandonprojects**	Out Of Total Consider Projects
1	2001	2005	SDLC	10	301
2	2006	2010	SDLC	18	
3	2011	2014	SDLC	28	

Table- 3: Overall Abandon Projects

Following **Graph-3** Shows The Same Portrait Depends On The Above **Table-3**. On X-Axis We Defined The Same Considered Years And On Y-Axis Abandon Projects Out Of Total Considered Projects



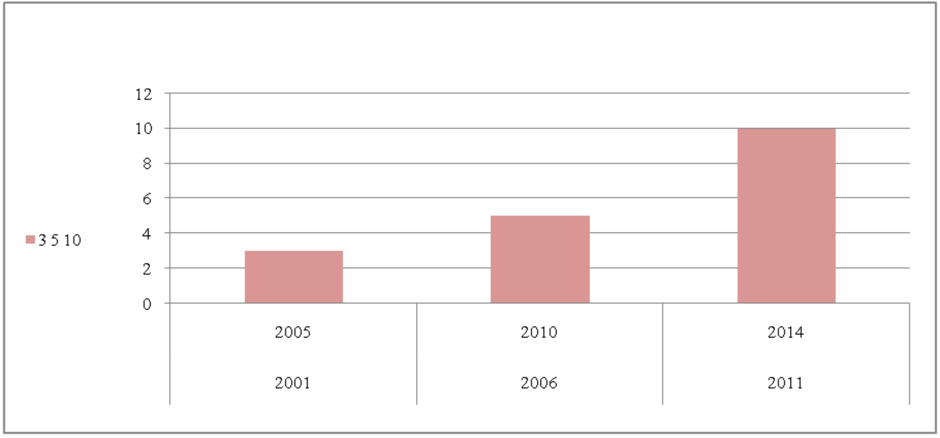
Graph-3 : Total Abandon Projects Out Of 301

Table-4 Shows That The Projects Are Fail During The Year 2001 To 2014 Out Of 301 Overall Projects.

Sr. No	Considered Year		Development Type	**Fail Projects**	Out Of Total Consider Projects
1	2001	2005	SDLC	03	301
2	2006	2010	SDLC	05	
3	2011	2014	SDLC	10	

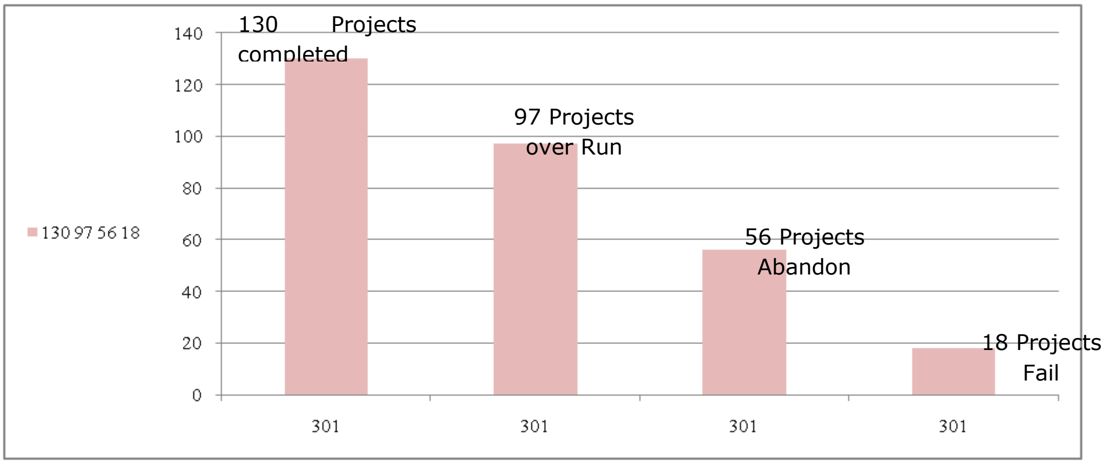
Table-4: Overall Fail Projects

Following **Graph-4** Shows The Same Portrait Depends On The Above **Table-4**. On X-Axis We Defined The Same Considered Years And On Y-Axis Fail Projects Out Of Total Considered Projects.



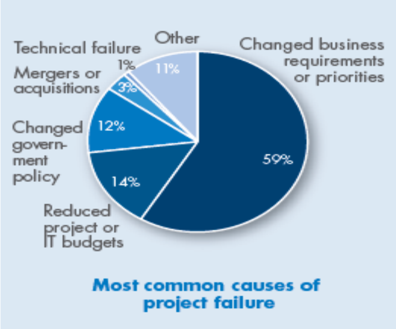
Graph-4 : Total Fail Projects Out Of 301

Graph-5 Shows The Overall Number Of Completed Projects, Number Of Over Run Projects, Number Of Abandon Projects And Finally Number Of Fail Projects Out Of Considered 301 Projects.



Graph-5 : Total Completed, Overrun, Abandon, Fail Projects

B) **CAUSES OF FAILURE:** Poor Requirements Engineering Is Blamed For The Majority Of Project Failures, According To A Common Complaint. In Fact, Requirements Issues Surpassed All Other Failure Factors Combined. Changing Government And Regulatory Regulations Have Been A Growing Role In Project Failure Over The Last 12 Months. [Adopted From 2]



Our Findings: We Can Able To Divide These Issues In To Two Major Portions A) Related With The User B) Related To Software Team

A) **Related With The User**

1) Continuously Changing Requirements.

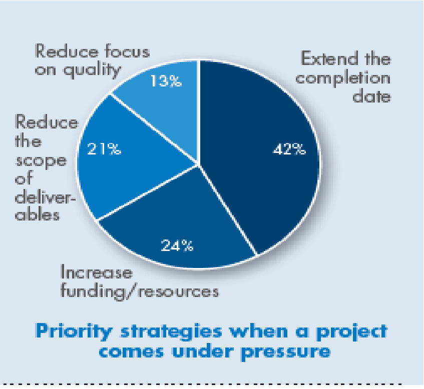
2) Not Properly Follow The Financial Agreements.

3) Not Cooperate Well With The Software Team Members.

B) **Related To Software Team**

1) 1) The Design Team Was Overconfident In Their Acceptance Of Specifications, And Developers Got Obsessed On Meeting Deadlines, Allowing Errors To Slip Through The Cracks.
2) Allowing The Software Testers Late In The Project.
3) Allowing Software Developers To Do Most Of The Testing, Rather Than Focusing On The Development Most Of The Time.
4) Many Businesses Have Encountered Accounting System Failures As A Result Of Software Flaws. The Problems Ranged From Incorrect Information Being Produced To The Entire System Crashing.
5) Insufficient Time
6) Inadequate Knowledge
7) Inappropriate Motivations
8) Insufficient Dedication.

C) WHEN PROJECTS COME UNDER PRESSURE: While Two-Thirds Of Companies Did Not Want To Compromise The Quality Of Their Original Goods, There Was A Considerable Increase In The Number Of Companies That Would Rather Lower The Scope Of Deliverables To Solve Project Problems. Despite The Evident Focus On Costs In Software Development, The Proportion Of Respondents Who Said Their Business Would Initially Boost Funding And Resources Increased From 20% In 2010 To 35% This Year.[Adopted From 2]



This Situation Is Always Be There In The Software Companies. Since To Meet The Commitment Of User Is Not Like Watching T.V. It Much More Difficult. For Gaining The Project From Your Competitor's Sometime Higher Authority Say Yes To All. But At The Time Of Implementation It Is Not That Easy, So Every Member Come With The Pressure Time In The Company. Experience Persons Handle This Situation Quite Well But Juniors Affect By This Situation.

Conclusion: Many Companies Including Software Company Mostly Relay On Customer Satisfaction As Well As Final Product. Final Product Need To Be Delivered On Time And Must Not Compromised With The Quality. So This Paper Focuses On The Same Outcome How Many Software Projects Are Going To Be Complete On Time With Real Cost, Overrun Projects With Re Cost To Be Considered, Some Abandon Project As Well As Failure Projects. The Software Development Model And The Model Are Also Discussed In This Work Based Try To Find The Bug Percentage Involved. Here The Bug Percentage Is Too High Because Of SDLC But If More Advanced Models Are Used We Can Reduced The Bug Percentage As Well. Next Shifting Our Focus On If The Software Companies Not Meeting The Deadline Then Find Out Re-Cost Valve Involved In The Projects. Next Considered The Ratio Related With Abandon & Failure Projects From 2001 To 2014.

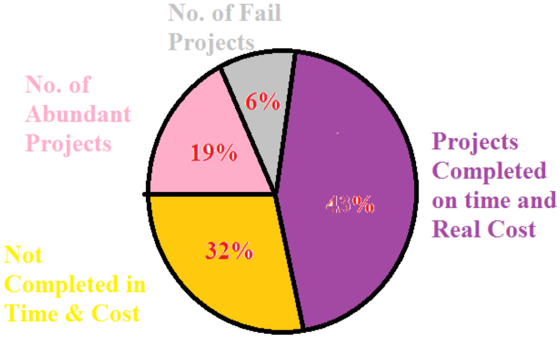


Fig. Projects Outcomes Of Last 14 Years (From Year 2001 To 2014)

Finally The Overall Percentage About The Project Completed On Given Time And Real Cost (43%), Projects Not Completed In Given Time And Cost (32%), Abandon Projects (19%), And Fail Projects (6%). Is Somewhat Similar To The Data Related To Reference Paper 2.

Sr. No	Consider Status	Reference Data	Our Findings
1	Projects Completed On Time With Real Cost	41%	**43%**
2	Projects Not Completed On Time & Re Cost	20%	**32%**
3	Number Of Abandon Projects	26%	**19%**
4	Number Of Fail Projects	4%	**6%**

The Reference Data Considered Is Provided In [2]. The Considered Data Is Only During The 2010 To 2011, But Our Considered Years Were From 2001 To 2014 So There Is A Little Change In The Outcome.

ACKNOWLEDGMENT: I'd Want To Express My Gratitude To All Anonymous Reviewers For Their Insightful Remarks, Which Were Used To Improve This Paper. I Am Grateful To Dr. Suryakant B. Thorat And Dr. Pradeep K. Butey For Providing Me The Appropriate Guidance And Assistance. Finally, But Certainly Not Least, My Family, All Of My Mentors Since Childhood, And My Friends For Their Unwavering Support.

References:

1. Introduction To Software Engineering
2. **Planittesting Index 2011 Australia &New Zealand** The **Benchmark** In **Software Testing And Systemsassurance**
3. “Engaging Testers Early And Throughout The SDLC Includes Seven Model” By Mark L. Gillenson, Xihui Zhang, Sandra Richardson.
4. Finding And Fixing Problems Early:A Perspective-Based Approach To Requirements And Design Inspections By Dr. Forrest Shull And Dr. Ioanarus , Dr. Jeffrey C. Carver
5. Measuring The Software Product Quality During The Software Development Life-Cycle: An International Organization For Standardization Standards Perspective Byrafa E. Al-Qutais (Journal Of Computer Science 5: 392-397, 2009 ISSN 1549-3636 © 2009 Science Publications)
6. Improving The Software Development Process Using Testability Research
7. Author: Jeffrey M. Voas Keith W. Miller
8. Importance Of Testing In Software Development Life Cycle Author: T.Rajani Devi (International Journal Of Scientific & Engineering Research Volume 3, Issue 5, May-2012 1 ISSN 2229-5518)
9. Testing And Software Technical Risk Assessments Author: Brad Neal, Simventions
10. Development Of Decision Models For Best Use Of Software Testing Resourcesauthor: Charles J. Campbell, Judith C. Simon, Ronald B. Wilkes