

STeP-IN SUMMIT 2008

5th International Conference On
Software Testing

A Structured Approach to Software Support Management

by
Alok Sawhney

Copyright: STeP-IN Forum and Quality Solutions for Information Technology Pvt. Ltd.

Published with permission for restricted use in STeP-IN SUMMIT 2008 in agreement with full copyrights from owner(s) / author(s) of material. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior consent of the owner(s) / author(s). This edition is manufactured in India and is authorized for distribution only during STeP-IN SUMMIT 2008 as per the applicable conditions.

Practices Experience Knowledge Automation

Produced By

STeP-IN
Forum

www.stepinforum.org

Hosted By



www.qstglobal.com

Introduction :

Almost all of software management processes primarily focused around developing and testing of new components or enhancements to existing ones. Methodologies on how to balance the development and test efforts are available and metrics can be derived and interpreted. However a significant part of the software budget is spent on maintenance, providing enhancements and fixes for reported problems. But there aren't any methodologies that address this phase of the software lifecycle in any great depth.

For any software product development company supporting the released versions of the software is as important, if not more, as working on the newer releases. Customer satisfaction and the long term success of software is as much a function of the quality and timeliness of support received as the features in the forth coming releases. If not managed right, the support effort could end-up taking the entire available development effort without leaving anytime for new development and enhancements.

Our SP process is a well planned defects fixing and release process as per a pre scheduled calendar for the year. Our whole Service Pack process being customer focused and proactive in nature, usually has 4 ~ 6 releases per year. While Service Pack's (SP's) no longer contain code for additional or improved features, they do (and will continue to) contain hotfixes (roll forwarded) that have been properly tested and fixes for commonly reported critical problems both internally and by the customers.

In here we propose a structured and efficient approach to managing support for release software versions, by means of Service Packs (SP) and HotFixes (HF). With this approach our organization has ensured that the customer receives timely and quality support with almost zero impact to the ongoing efforts on developing new features and enhancements. Hotfixes (HF's) can also be termed as updates, interim updates or Quick Fix Engineering (QFE). The purposes of HF's are to provide a quick fix for the organization that have discovered issues between SP releases. Service packs correct known problems and provides tools, drivers, and updates that extend & improved product functionality, including enhancements, if required.

Software Support Management definition

Our software support management process includes the following types of code introduction/ modification and improvement.

- **Service Packs:** These are incremental releases of the baselines i.e. each new service pack contains all the fixes in previous SP's, as well as any new fixes of HF's (roll forwarded), critical updates and other updates.
 - Hotfixes: All Customer Hot Fixes are rolled into the next available Service Pack release.
 - Critical Update: This is not a software upgrade but a fix for an issue in broadly applied software. These are mostly customer reported issues but not show stoppers.

- Internal Update: This addresses important, non-security issues, but mostly internally reported ones, identified after the release of the main software.
- Feature Pack: This includes non-critical additions to the base software program. It typically appears between major releases for the important issues, found internally, late during our regular testing process.
- **Hotfixes:** Patches built to address specific issues. The Customer Hot Fix Process will handle ad hoc defect fixes required for customers that have reported show-stopper defects – P1/ S1 defects. Defect fixes are turned around at the earliest opportunity based on a structured approach to defect fixing and resolution testing.

Why the need for Support Management - Typical Key Challenges:

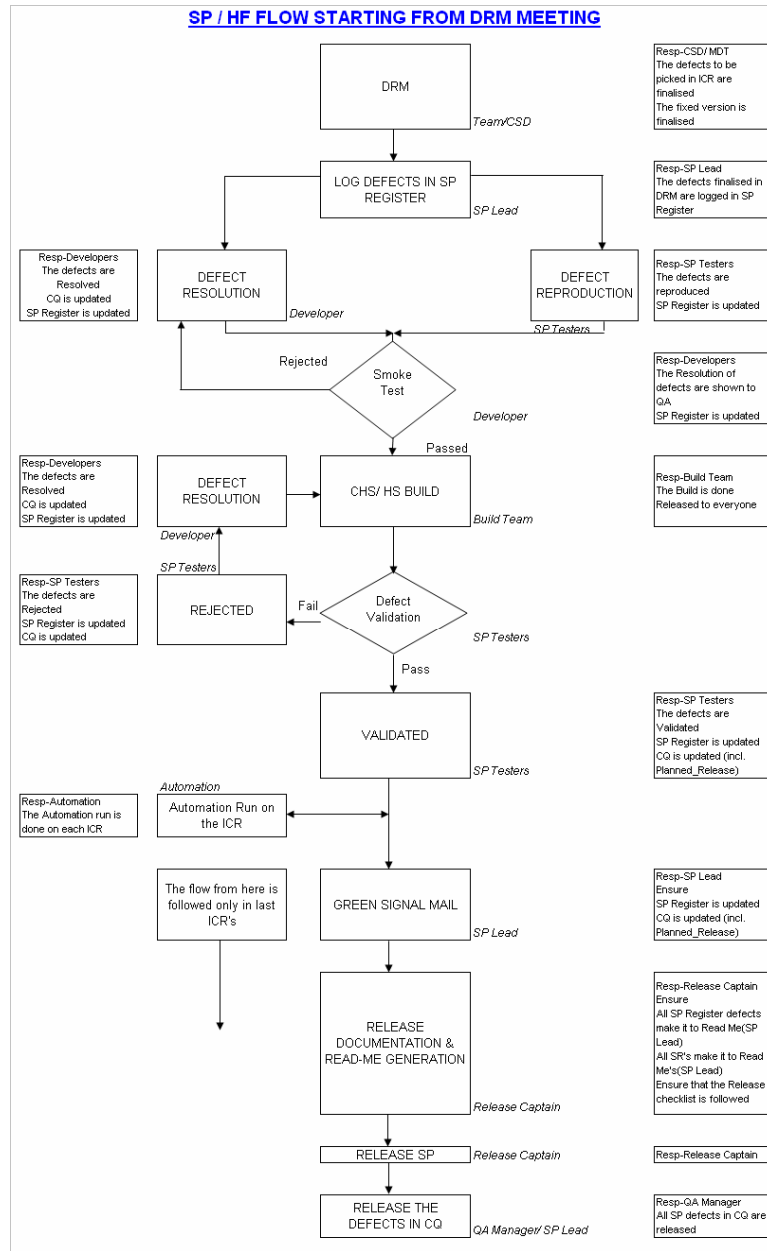
In spite of having all the best practices of software development in place, the key challenges in supporting released software are -

- Is there a best time for service packs?
- Not all defects can be found and fixed before releasing any non-trivial software - How does one ensure that the release is timely and also that no critical defects are left out?
- Not all defects require a huge amount of effort to fix, but once released the effort required to integrate and system test is large. How can the required development capacity still be balanced with the required test capacity?
- How to address related defects found during defect validation – as a result of bad fix?
- How can risks be assessed, enabling a risk-based test strategy?

SS Management- Approach adapted:

These challenges are addressed in our existing software support management process as-

- Yearly SP release schedule is planned in advance and published to customers. The SP release dates are planned in such a manner that they fall in-between two major releases. One SP release is planned immediately after the main release so that any left over fixes which could not be included in the main release or any last minute major issues identified gets included in the SP.
- Even though HF's are usually not pre-planned, still if any show stopper issues are identified internally, which may hamper a customers flow, are released in planned HF release. The complete yearly schedule (as shown in appendix-I below for two iterations) for the SP are released to the customer in advance so that they can plan their internal activities accordingly.
- Balance between the development and testing capacity is maintained by having dedicated resources allocated for the SP testing. The engineering manpower is pooled and allocated from the regular process, which is pre-planned and already considered for planning main releases.
- Dividing a SP into multiple iterations so as to be able to have incremental releases made to the customer, if required. The turnaround time gets shortened due to the parallel execution of engineering and testing activities.
 - Rejections due to bad fixes, taken up in the next iteration – ensuring timely release of the SP.



The typical process flow for Service Pack and HotFix patch management is-

- Planning and defect short-listing/identifying
- Kick off / Review meeting overview
- Individual checking / defect recreations- manually & creating automation scripts/ environment set-up
- Root cause analysis initiation
- Engineering activity / unit tests / developers review
- Smoke test
- QA testing / validation- manual and automated scripts executions, exploratory & regression testing.

- Rework, if any rejections or shortcomings identified
- READ-Me's generations (supplied along with the software with installation details and defects fixed details), defect tracking software update and final GREEN Signal from QA for initiation of the software release process.
- Software release activities including prior intimation to customers of the details of the fixes made.
- Software final release

The four phase approach adapted is-

Phase I- Iteration Plan and Defect selection

- The entire process of service pack defect selection and its inclusion are governed by customer's requirement and their expectation. Customer satisfaction being one of the important criteria within the organization is well taken care by this effective SS Management process.
- All the defects finalized to be included within a SP are logged into our system using adequate tools for further tracking through the complete journey of Service Pack and HotFix releases. Once finalized, customers are informed about this along with the already planned release date. The defects selection, its risk, priority and its categorization whether it's a defect or an enhancement is decided in the support clinic attended jointly by representatives of CSD, Marketing, Engineering and QA. These defects are then included by CSD in a work plan for the forthcoming SP iteration.

Phase II – Elaborate and Review –review proposed solution, analyze impact and environment setting.

- The work plan for the particular iteration is discussed and agreed in the review meetings. All corresponding impacted areas identified and also included in the fix. Multi disciplinary team of all departments concerned work together in reviewing both customer and internal defects and prioritizing the ones to be fixed. Depending upon the resources available, the criticalness & severity of the defects and estimation of the work involved in its fix & testing, each defect is discussed and agreed in the review meeting to be included /postponed or rejected for the forthcoming planned SP. Chaired by Customer Support, this review meeting-
 - Discusses and analyze the planned fix for the defect included in the Service Pack/ Hot Fix Work Plan.
 - Prioritizes the defects based on the criticalness, severity, complexity of the defect and the estimated time to fix and test the defect.
 - Updates the Service Pack/ Hot Fix Work Plan accordingly.
 - Approves the Service Pack/ Hot Fix Work Plan and SP register updated accordingly. The finalized defects are then updated in the SP register (as shown below) by the QA team. This SP register is then used as a matrix to check & control the status of the SP/HF in future.

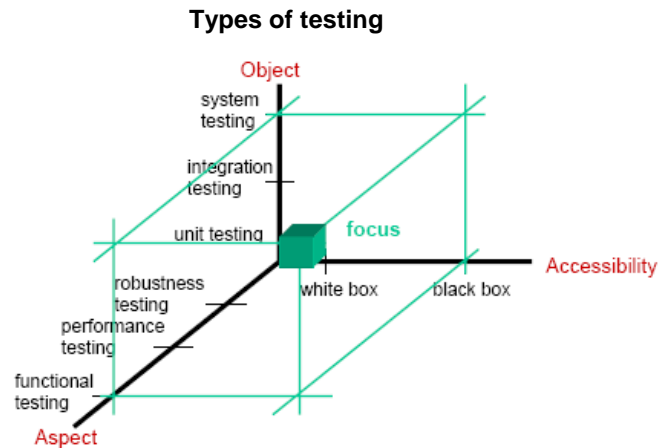
SP/HF Register-

- QA fills in the testing that was done
 - Root cause of why problem was not found during testing
 - Propose Solutions in RCA document
 - Root Cause Documents Reviewed
 - Invite Engineering & QA involved in the concerned SP/ HF defects over the last cycle.
- Replicating customer environment, to the extent possible.
- Creating the replica of customer environment is the foremost requirement for successful testing process for SP and HF defects. Ideally, one should be able to duplicate all of the various system types and implementations customers or the end users have in their environment (all applications, operating systems, and combinations thereof). However, in the real world, this is unrealistic for most organizations.

Instead, for a test environment, we create the systems that represent most of the technologies (if not in all their incarnations) used in production. For example, if the customer uses the Apache web server on Linux, we would have a representative server using these technologies (not necessarily mirroring all content and applications). One would have representative desktop systems loaded with common business applications. To make this even simpler, we use a virtualization tool like VMWare to create multiple operating system images that can run simultaneously on the same machine.

Phase III - Development and Testing – develop and review solution, validation.

- Development and defect maintenance
- Each SP & HF released after the main released version is baseline using source code version control tool, there-by making the SP's and HF releases being incremental releases of the baselines. After the SP/HF has been released the same modifications are rolled forwarded into the main release source code available for the next release candidate.
 - This process helps in parallel development activity being done in both the main stream release codes as well as the SP or pre released versions codes.
- Once our environment is reasonably composed as per the customers replica environment (as far as possible), we work out on elaborate steps to reproduce the actual defect using the same. The testing type & its process depend on each individual defect risk, priority, severity etc.



The basic SP testing process is based on the following approach

- **Unit Testing and Code review by the developers** - The unit tests are prepared after due careful consideration of the defect changes and its surrounding affected areas by the developers. The following are the processes used for unit testing:-
 - Unit Test Checklist

For GUI, there is a unit test checklist that covers the appearance and usability of different GUI objects i.e. checkboxes, radio buttons.

For non-GUI code, the most effective method of unit testing is via automated unit tests. Wrapper java classes are created that are designed to call the “real” java classes with varying parameters with the objective of exercising as many lines of code as possible.
 - Automated Unit Tests- Individual Java classes are executed by Java test classes. Unit Tests are executed on developer code base at all the development sites on a daily basis.
 - Static Code Analysis- IntelliJ is able to examine Java code for “chosen” problems without executing the same.

The peer review of codes is conducted so as to catch the possible coding problems, if any. This is basically done so as to find defects, educate, promote best practice, ensure good code and identify redundant codes.
- **Smoke test done on development build** - to review the fix in advance and reduce/ avoid rework due to rejections at later stage. The areas fixed and impacted due to the defects fix is shown to the Marketing, QA & Technical Authors so that problems are noted/ fixed before the kit is release to QA for formal testing.
- **Defect Recreation** – done manually.

All the defects included to be fixed in any incremental builds are first recreated in prior released build so that the actual behavior of the defect is noted. This recreation is presently done manually as per the steps to reproduce the same, by the QA tester.

- **Defect Validation** – manual testing as per the steps to reproduce and defect flow.

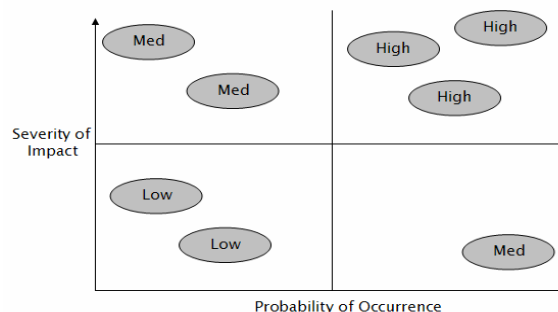
First, we want to ensure the actual installation and deployment of the SP / HF patch works on our representative systems, which gets validated as part of our testing process. Once we have tested the installation of the patch, we can proceed to user and application testing. This consists of normal operating tests to ensure that the system and applications continue to work as expected post-patch.

Depending on the amount of testing required for a particular defect, a number of specification techniques are allocated. For higher risk defects, multiple techniques are involved. Faults with most rework and high priority faults identified early. A risk assessment is done on the identified features to enable them to be sorted into a priority order. Two things are considered during this assessment:

- Severity – estimated by Marketing
- Probability – estimated by Engineering

The aim of the risk assessment is to identify the high risk features so that an order to the testing of the features can be established, with the high risk feature tested first. As shown in the diagram below, the aim of the risk assessment is to identify the features that fall into the High category. Identifying the Low priority features is also very useful for this process.

Risk Based Analysis- Likelihood/ impact diagram



- **Exploratory testing** - find any new defects introduced as a result of code changes. All the surrounding areas affected by the code change done for the SP defect's fix, are covered manually.
- For any defect being rejected due to improper fix implemented, the same is carried forward to the next iteration, if agreed upon by all the concerned departments, depending upon the defects severity and criticalness of the fix implementation to the customer.

Phase IV – Transition and Release - Regression testing, deliver solution, defect tracking, release process.

- System/ Regression Test – regression checklist followed, automation suite covering previously fixed defects and process flows are executed. Our comprehensive Automation suite is executed on each builds so as to check for any regressions for the previously fixed defects and software behavior for flow executions. Automation has become the part and parcel of testing because quality assurance is considered one of the most time consuming phase of the product cycle. Automation is all the more important for service pack releases as we need to run the same defect in different customer's environments and different databases installations and operating systems. The automation suites are used for several purposes:
 - Used to test hotfix and service pack releases (executables)
 - Used for regression testing in new releases (executables)
 - Used for regression testing (the developer code base) directly within the development environment.

In addition to the normal manual testing for each defects, these automated regression testing are executed for each and every SP and HF's before sending the GREEN Signal for its final go-ahead for the release process initiation. The GREEN signal is a confirmation from the QA side that all the necessary testing and validation of the concerned SP/HF has been completed and the subsequent release activities can be initiated henceforth.

- Multi-platform testing on the customer replica environment (as far as possible) is done for different OS, databases etc. supported by the product.
- Once the SP/HF testing process is completed, the tester automate the selected few defects so as to add on to the existing automation suite already available and executed before sending the final go ahead for the release activity.
- Web based tracking process followed for the customer to track the status online and the fix details of their defect, at any given point of time. The Release Contents are updated at the Mentor website for all the customers to view the same. A list of all resolved defects (core and kits) is added in the "What's New" section of this form.
- Automatic notification sent to customer once the defect is fixed, by the defect tracking system
 - The core success of this whole process is realized by timely and adequate tracking and release process being followed within the organization.
 - Review README file is supplied by QA for each version of the Service Pack's. The README.htm files include the following:
 - Current copyright header
 - SR number, Product (CHS only), Headline and Customer note extracted directly from CQ
 - Installation instructions and baseline version

- Any additional notes
- The whole Service Pack and Hot fix release process is governed by “SP/HF Release Checklist” as shown below.

Release Captain – SP/HF Release Checklist

Task	Complete?
Register Patches	
Email Release Analyst with planned release date	
Email to all concerned to sign off MRCs	

Prior to release:

Task	Complete?
Create README.htm files	
Get README files reviewed	
Upload README.htm files to Internal FTP server	
Complete Minimum Release Criteria (MRC) Checklist	
Email MRC checklist to concerned for final approval	
Add list of resolved defects to patch registration form	
Zip up SP's	
Create ISO files for VCD	
Upload zips/isos/checksums to WOF FTP server	
Submit Posting on Intraware form	
Email WOF with approved MRCs	
Check test links and email approval to WOF	
Check live links	
SP emails	
Update Release Kits area	
Email to concerned for updating the “shipped” field in CQ	

Hotfix Management Process:

Overall the HotFix management is similar to that of the SP process except for the following few changes due to the short turnaround time for HF process, starting from the customers intimation and right upto the end software release with the adequate fix.

- **Hot Fix Work Plan-** The Hot Fix Work Plan will document the defect(s) that are stopping a customer. It also records the defect's resolution complexity rating and an estimate for the time to resolve the problem.
- **Customer Hot Fix Review Meeting-** The Customer Hot Fix Review meeting is held within a day of the defect being reported to the Customer Support Helpdesk.
- **Test Resolution-** A full impact analysis is performed and the appropriate corrective actions are implemented for the impacted areas of functionality & check list updated by the developer.
The defect resolution and its associated impact are unit tested on the developer's workstation. Where appropriate, a code coverage tool is used to verify that all modified code and functionality is fully tested, i.e. all modified code is executed by the unit test. The defect resolution is transferred onto the Integration Test workstation and integration testing is performed.
- **Test Scripts-** As part of the defect reproduction and test process any automated or manual test scripts created are made available for testing the resolution as part of the regression test of the defect
- **Customer Hot Fix Release-** Customer Hot Fix Releases will be the final deliverable of executables and libraries that are sent directly by Customer Support to customers.

Key Measures:

The key measures used to determine the progress of the SS Management process is

- Team/ Individual staff weekly progress report.
- Post SP release, each defect is analyzed in terms of the
 - Trend of the defect type & product wise within the current SP and previous SP releases.
 - Trend for customer wise defects reported.
 - Internal re-builds/ rejections required per iterations due to improper fix.
 - Identifying further areas of improvement by means of RCA analysis & post release analysis.

Key Benefits:

- Effective and timely release of SP's, as planned.
- Customer satisfaction is at its best; in addition to getting their issues fixed, they can also plan their related activities
- Reduction in cost for the organization by effective utilization of development and testing teams.
- The RCA process has helped in identifying certain key changes to existing processes, testing methodology, identifying key areas for additional automation, extra unit tests, etc. This has in turn helped in getting fewer HF's in future and also improving product quality.
- Some critical defects found late during the regular testing process which cannot be included in the main software release kits, gets fixed and delivered to the customer via the next available SP's.

Key Success factors:

Experience shows that following are the key success factors of the whole Support Management process-

- Effective and timely release of planned SP & HF's.
- Adequate measures in place for effective tracking of the defect status to both internal and external customers.
- Effective and exhaustive automation process in place so as to cover the maximum possible regression issues and multiple execution of the same defect in different versions & installations.
- Effective RCA process in place so as to understand the root cause of the defect identified by the customer, on the released software version.
- Effective feedback process back to the main stream testing so as to improve upon the exploratory checklist & testing process.

Road Ahead:

We are presently working upon to improve further and add upon our existing automation suite so as to cover as much possible flows and previously fixed defects. This will definitely not only strengthen upon our existing regression and stress testing process, but also improve upon our defect validation process.

As per our existing process, the defects are presently being recreated manually in the phase II of our process. We are working towards the process where-in the defects will be recreated by automating the same at this stage only. The intent is to have these scripts being failed, when executed on the build prior the fix has been implemented. Once the SP/HF fixed build is received, as part of the defect validation process in phase III, we run these automated scripts. The automation script previously created while recreating the defect should now get passed after the fix has been implemented.

Conclusion

In summary smart work and not just the hard work is needed to reduce the cost of testing, thereby having efficient testing activities in the software business.

As the SP & HF fixes are built according to most of the expectations of the customer, the customer satisfaction level is high. This is the most important outcome from the overall process of effective and timely release of planned Service Pack and HotFix releases.

Profile:

Alok Sawhney has approx 13 yrs of experience working in diversified areas and different companies, of which 8 years of experience is related to design engineering, product, process engineering and quality system standard implementations in the automotive sector and 5 years of experience in IT service, QA, CAD tools, Six Sigma process, Business Excellence modules, QA Software Testing and Test automation processes.

He was associated with Motherson Sumi Systems Ltd. for approx 8 years and then worked for 4 yrs. in Tata AutoComp Systems Ltd. in its Engineering division. Responsibility wise, he have successfully managed the teams consisting from size, varying from 5 ~ 23 members. Have mentored and trained fresh engineers about the processes and best practices within the industry. Having the domain expertise in electrical system (wiring harness) designing and testing areas, he has been associated with Mentor Graphics (India) Ltd., since last 1.5 years in their Integrated Electrical Systems Division's testing team.

As a certified software test professional by ISTQB/ISEB, he has been involved in test management and life cycle testing of software products, using different test techniques. Starting with test preparation and execution for one of the IESD tools, he is presently involved in the effective implementation and continuous improvement of the software support management process within the organization as a SP/HF software tester. He has been quite successful in providing his value added services in regular release and SP testing process improvement, automation process implementation etc.

Alok has done his Bachelors in Electrical Engineering and then completed his MBA in Operations Management. He has a good exposure to QS 9000, VDA 6 and ISO 9000 Quality standard requirements for Quality Management System and Internal Auditing. He has also been associated in Business Excellence Module implementation & auditing, in line with the Malcolm Balridge National Quality Program (US)