



The Synergy Approach™

Raghav S. Nandyal
Chief Executive Officer

SITARA Technologies Pvt. Ltd.
'Bellevue', #559 • Road No. 3 • Banjara Hills • Hyderabad AP - 500 034 • INDIA
Email: raghav_nandyal@SITARATECH.com
URL: <http://www.SITARATECH.com>

A Near Zero-Defect Approach to Software Development

From The

Software Engineering Laboratory





Foreword

Lectures on Software Engineering as delivered by Raghav Nandyal during his consulting engagements worldwide are being presented as a series of Technical Reports. These lectures from SITARA Process Jewelbox™ serve as good introductory and authoritative reference material on Software Engineering from a person who has won several awards for his effort on **"Process Automation"** from Motorola during his engagements with Motorola. Raghav Nandyal is an internationally recognized expert in areas of Software Engineering and serves on the review panel for IEEE Software.

The Software Engineering Laboratory at SITARA Tech is delighted to present these lectures. The tone of the lecture series is spoken style, and where editorial intervention was thought necessary, they have been made. If you need more information or would like to provide feedback, please contact-

Shailaja R. Nandyal
Management Consultant
SITARA Relationship Management Team
Email: shailaja_nandyal@SITARATECH.com
URL : www.SITARATECH.com



March 1995

For a software development initiative to qualify as a “discipline”, activities that can be considered neutral to who executes it but depends upon a methodical process of execution is required to be established. Such a process would have well articulated set of activities that govern development and subsequent maintenance of software. You not only document what you do but do what is documented as – the process. Such a process that is broken up into manageable chunks of execution, called phases, would describe those activities that will be performed on well-defined inputs after verifying the entry criteria. Deliverables at the end of each phase would then be archived after reviewing to detect any errors. The development process found in the ensuing lectures are **NOT** intended to be a software life cycle of its own. Experiences with existing best practices coupled with practical needs from the many development projects that were undertaken have been the major inputs in defining an illustrative process. The reason for calling it the “**Synergy Approach**™” is simple: I perceive a need -

“To provide a humanizing element to a disciplined software development process that encourages spirit of teamwork and interdependence while emphasizing automation of transitions across phases”.

To answer, “How is the Synergy Approach different from the Waterfall Model, V Model and the Spiral Model?” - it would be necessary to review these models against the Synergy Approach.

In the waterfall, you sequentially execute phases one after another by “falling through” the lifecycle of development. Synergy Approach accommodates the real life situation of continuous change that the Waterfall Model cannot address. By doing away with separate verification and validation activities of the V model, it exploits synergy of the team to build robust solutions. By having the entire team, both development and testing provide perspectives based reviews of each phase of the life cycle to assist in harmonizing the different activities and the deliverables, it reduces the need to iterate among phases that is addressed by the spiral model – but when poorly managed, can lead to the death spiral! In providing all these advantages, the Synergy Approach improves the “quality” of the end of phase deliverables by doing the right thing right the first time. And unless you follow the project team structure that is suggested later in these lectures on Technical Pool Organization, you will not be able to get it right.



1.0 Introduction

The illustrative lifecycle that will be used for the presentations and lectures are a "common-sense" approach to breaking down complexity. The governing idea behind the **Synergy Approach** is to build a quality product with the help of software development professionals who are motivated by the spirit of "**interdependence and team-work**". As the competition to excel as a premier software company heats up, it becomes very important to deliver a quality product (near zero defect software), which has in it all the virtues to please not just the customer but also the end-user, besides being delivered on time meeting the market window of the product. The only hope available for us to meet these demands is to exploit the synergistic outcome of putting to use the very best available in each one of us and to build teams or pools of expertise and wisdom. And, these teams will be geared towards interdependence and teamwork while emphasizing "**automation**" of as much of the tasks involved in achieving the goal of producing "**quality software products**". These teams will also be responsible to create repositories or reuse libraries of all the automated solutions emerging from their group. One of the key elements in such an effort is "**effective communication**" among the groups or pools of expertise. This can happen if the end-of-phase deliverables or configuration items are well documented in a consistent fashion so as to be easily understood by peer reviewers.

Reviews are quality checkpoints where errors are trapped. All configuration items shall be reviewed until all *known* defects are zero. A "defect" is defined as "errors" that have escaped a phase into the next one. Stated differently, "Any undetected error is a defect".

2.0 The Synergy Approach™

Keeping psychologist George Miller's famous magic number of 7 ± 2 linear simplification of a complex process, this approach is made up of six phases culminating with the final customer approval. Each phase will have end-of-phase deliverables that will go through a formal peer review. And the modus operandi to conduct formal peer reviews will be described in a separate presentation.

Reviews of each of the end of phase deliverable are conducted to weed out errors and defects from the configuration items. And, customer involvement in as many of the end of phase reviews as possible is a very desirable step and should be scheduled in the project plan. A configuration item is a peer-reviewed work product that would be archived at the end of each phase. If there are changes to the configuration item as a

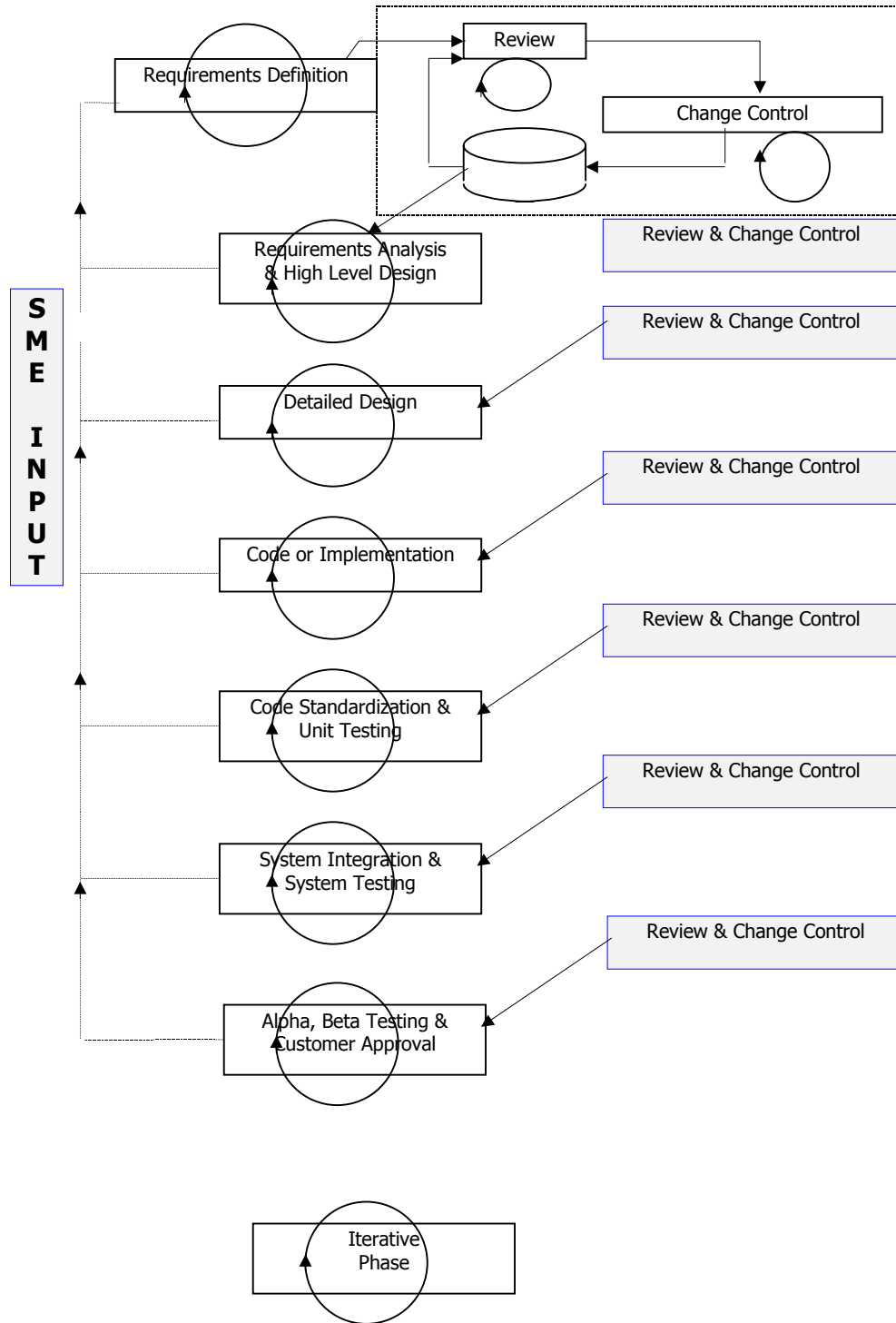


result of changed customer requirements, a well-defined version control mechanism that would be under the control of the "software configuration manager" (**SCM**) or by the project manager himself (**PM**) – if it is a small team, will have to be undertaken. The software configuration manager of a software development team can also be the technical lead of the project, also called as the software systems engineer (**SSE**). Note that in the best-executed projects, these are role-plays and not titles that people carry.

The Synergy Approach™ identifies an illustrative process to comprise of the following phases. The reason it is called an illustrative process is because, tailoring must be exercised as an option.

1. Requirements Definition Phase
2. Requirements Analysis and High Level Design Phase
3. Detailed Design Phase
4. Coding or Implementation Phase
5. Code Standardization and Unit Testing Phase
6. System Integration and System Testing Phase
7. $\alpha\beta$ Testing and Customer Approval Phase

This approach is illustrated in the figure.





It is very important to note that the spirit and purpose of the synergy approach is very different from the commonly used models. In both the Waterfall and the V models, two separate teams conduct the development and the testing activities "independently". This approach does not augur well in practice. Oftentimes, with such independence governed by the principal motivating factor for the testing team coming from - "a test case must break the system", leads to needless conflicts with the production of near zero defect software with emphasis on reduced cycle times. This is because; the amount of quality and robustness that can be built into a product in the software system-testing phase is expensive and extremely difficult if not impossible. Imagine the consequences of having to redo the design as a consequence of system test reports that gets generated almost when the product has to be shipped! The promotion of the thought of an "independent product development and test team" does not hold in a zero defect software development environment.

In reality, one must question the philosophy governing such an approach namely, "the test team will develop system test cases independent from the development team and will subject the product with these cases till it breaks". The intention is good, "to think of situations that the software will operate in and come up with test cases that *might* have been overlooked by the development team", but the execution will almost always be poor. The goal of the test team is most definitely not to point out mistakes but to help to correct them and think "with" the development team of the various scenarios that software solutions may encounter. If zero defect software is the goal, then the test team must work in union, in a synergistic way to help the development team build quality into the product design. Besides, nothing much can be gained from having the mistake being pointed out during the system-testing phase when it could have been eliminated much earlier. The point that is being made is, "a stitch in time saves nine" and execution of each phase must be approached by the "team" in an interdependent fashion (NOTE: the identity of a development team that is separate from the test team does NOT exist in the Synergy Approach™).

END OF LECTURE 1

CONTINUED ON LECTURE 2 [SITARA SE/JUNE 2001]