CHAPTER 6

CHAOS: THE COMMON CAUSE OF DATA WAREHOUSING PROJECT FAILURES

Chaos is one of the oldest concepts and paradigms. Its origin could be traced back at least to Greek mythology. In this framework, Chaos was the disordered world that preceded Cosmos that is our present ordered world. Therefore, in its wide sense, chaos merely refers to some kind of disorder. And indeed, it has been used for rather distinct types of disorder. For instance, called "pure chaos" the Brownian motion, which he mathematically formalized. More recently, referred to "Levy chaos" as the extension of this pure chaos to motion, defined with the help of the Levy variables, which broadly generalize the Gaussian variables? Which chaos in the rainfall-runoff process. Deterministic chaos. During the last 30 years, (not clear-what is Deterministic chaos).

Chaos took a much more a restrictive meaning, since it became understood as a shorthand for "deterministic chaos". The latter denotes the disorder generated by deterministic dynamical systems. Here the adjective "deterministic" means that the equations do not contain any noise source, and that, in general (but not always), the existence and uniqueness of solutions are mathematically assured (Leticia et al. 2013). The main defining feature of such a chaotic system is the sensitive dependence of solutions on the initial conditions (and/or boundary conditions for partial differential systems). The prototype example is the celebrated Lorenz model, which was introduced as a mathematical caricature of atmospheric convection and has the lowest possible dimensionality, i.e. three, for chaotic differential systems.

6.1 Motivation for the Study:-

The Chaotic often a unsystematic work process in software development was seen as one of the main reason for the software crisis of the 1960s. Scientists from around the world sat together to discuss the software crisis at NATO software Engineering conference held in 1968, Garmisch Germany. Since then, Software Engineering has come a long way with a goal of incorporating engineering approaches in to the software development process.
Achievements towards this goal include a greater understanding of the role of abstraction and separation of concerns in Software Engineering (Lehman et al. 1984). The introduction of modularity and notations, etc. Most of these ideas come directly from engineering, although they need to be adapted to the unique problems that arise while working with different and more abstract materials (Haorianko and David 2005)

- Ad hoc requirements management.
- Ambiguous and imprecise communication.
- Brittle architectures.
- Overwhelming complexity.
- Undetected inconsistencies in requirements, designs, and implementations.
- Insufficient testing.
- Subjective project status assessment.
- Failure to attack risk.
- Uncontrolled change propagation.
- Insufficient automation.

### Table: 6.1 Root Failures of Projects

<table>
<thead>
<tr>
<th>Root Cause</th>
<th>Preventive Actions</th>
<th>Assigned to</th>
<th>Implementation Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standards not followed</td>
<td>Do a group reading of the standards all(after they have been updated)</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ensure that standards are followed in the mock Projects done</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard/Checklists (Incomplete attention)</td>
<td>Do a group of the review standards with expert from outside and then update the standards</td>
<td>XXXX</td>
<td></td>
</tr>
<tr>
<td>On sight Incomplete (Incomplete attention)</td>
<td>Effective Self reviews, Rigorous code reviews</td>
<td>All</td>
<td>Immediate effect</td>
</tr>
<tr>
<td>Unclear/incorrect Specifications</td>
<td>Specification reviews</td>
<td>All</td>
<td>Immediate effect</td>
</tr>
<tr>
<td>Lack of training</td>
<td>Every new entrant will do a mock project, Whose code will be reviewed and tested thoroughly</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To reduce the defect injection rate significantly, The Project manager decided to tackle the top three categories of defects: Logic, Coding Standards and the redundant code. A brainstorming session was held to identify the root causes and possible preventive actions. The regular procedure for brainstorming was followed. First, all the possible causes that any once suggested were listed, and then ones that were identified as the main culprits were separated out. For the causes, Possible

6.2. MULTIPLE CASES STUDY:

The objective of the multiple cases study was to collect data on the various aspects of the software development (Stapleton 1997). For convenience, the various aspects of the software development divided into three categories.

- People-related
- Process-related
- Product-related

Each of these aspects was further explored in detail to identify the independent variables that contribute to the chaos in software development. From the past experiences and literature survey conducted for the study, 26 independent variables. The GQM approach was selected to collect data on these independent variables. The approach has identified 1 goal, 25 questions and 132 metrics. Data on these 132 metrics form selected projects were collected by the multiple methods like specially designed question are, onsite observation, co-operation and project monitoring. The collected data from the
projects selected for the case study is used rank the 26 independent variable based on the severity of the chaos caused by them. Ranking of the independent variables was also influenced by the discussions and deliverable in financial year (Sam and Dennis 2003). Project Manager, Project leaders and developers. The 26 independent variables, their contribution and cumulative contributors are shown in the contribution table.

Table: 6.2 Contribution Table

Contribution Table: Factors contributing to chaos with the percentage of their individual and cumulative contributions.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Most Factors contributing to chaos in project development</th>
<th>Source</th>
<th>Contribution in (%)</th>
<th>Cumulative contribution in(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Incomplete Requirements or Project Delivered successfully Situations</td>
<td></td>
<td>10.50%</td>
<td>10.50%</td>
</tr>
<tr>
<td>2.</td>
<td>Documentation Completion failure and late delivery</td>
<td>Leon et al (2006)</td>
<td>8.50</td>
<td>19%</td>
</tr>
<tr>
<td>3.</td>
<td>Ineffective project planning and or management</td>
<td></td>
<td>8%</td>
<td>27%</td>
</tr>
<tr>
<td>4.</td>
<td>Unclear objectives</td>
<td></td>
<td>7.5%</td>
<td>34.5%</td>
</tr>
<tr>
<td>5.</td>
<td>Unrealistic timeframes</td>
<td></td>
<td>7%</td>
<td>41.5%</td>
</tr>
<tr>
<td>6.</td>
<td>Technology incompetence</td>
<td></td>
<td>7%</td>
<td>48.5%</td>
</tr>
<tr>
<td>7.</td>
<td>Communication breakdowns</td>
<td></td>
<td>6%</td>
<td>54.5%</td>
</tr>
<tr>
<td>8.</td>
<td>No or insufficient risk analysis documentation and process</td>
<td>Leon et al(2006)</td>
<td>6%</td>
<td>60.5%</td>
</tr>
<tr>
<td>9.</td>
<td>Changing requirements and specifications</td>
<td></td>
<td>6%</td>
<td>66.5%</td>
</tr>
<tr>
<td>10.</td>
<td>Project team members are overscheduled</td>
<td></td>
<td>5.5%</td>
<td>72.5%</td>
</tr>
<tr>
<td>11.</td>
<td>Failure in project manager Contributions</td>
<td></td>
<td>5.5%</td>
<td>77.5%</td>
</tr>
<tr>
<td>12.</td>
<td>commitment Level failure to project team</td>
<td>Leon et al (2006)</td>
<td>4.5%</td>
<td>82%</td>
</tr>
<tr>
<td>13.</td>
<td>Latest Technology</td>
<td></td>
<td>3%</td>
<td>85%</td>
</tr>
<tr>
<td>14.</td>
<td>No change control process (change management)</td>
<td></td>
<td>2.5%</td>
<td>87.5%</td>
</tr>
<tr>
<td>15.</td>
<td>Key project stakeholders do not participate or participate irregularly in major review meeting</td>
<td>Leon et al(2006)</td>
<td>2%</td>
<td>89.5%</td>
</tr>
<tr>
<td>16.</td>
<td>Poor cost and schedule estimation</td>
<td>Lorin(1998)</td>
<td>2%</td>
<td>91.5%</td>
</tr>
<tr>
<td>17.</td>
<td>Lack of user involvement</td>
<td>Lorin(1998)</td>
<td>2%</td>
<td>93.5%</td>
</tr>
<tr>
<td>18.</td>
<td>The lack of project methodology</td>
<td></td>
<td>1%</td>
<td>94.5%</td>
</tr>
<tr>
<td>19.</td>
<td>Unrealistic expectations</td>
<td>Steven(1996)</td>
<td>1%</td>
<td>95.5%</td>
</tr>
<tr>
<td></td>
<td>Lack of executive support</td>
<td>1%</td>
<td>96.5%</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---------------------------</td>
<td>----</td>
<td>------</td>
<td></td>
</tr>
</tbody>
</table>
| 21.| Lack of resources         | 0.05% | 97%
| 22.| Inadequate design         | Steven(1996) | 0.50% | 97.5%
| 23.| Difficulty in determining the input and output of the system | 0.50% | 98%
| 24.| Unstable organization environment (such as changes in senior management or restructuring) | 0.25% | 98.25%
| 25.| Poor architecture         | 0.25% | 98.5%
| 26.| No business case for the project | 0.25% | 98.75%
| 27.| Others                    | 1.25% | 100%

**Note:** The items for which no source is listed were not identified from earlier studies; They were added based on the authors’ experiences and on feedback from experts.

### 6.3. RISK ANALYSIS

A project chaos may be caused by the above problems. Having identified the hazards that might affect our project we need some way of assessing their importance. Some will be relatively unimportant (Ex: The risk that some of the documentation is delivered a day late). Whereas some will be quite likely to occur (it is quite likely, for example, that one of the software developers in a team will take a few days Sick leave during a lengthy project). Whereas others are relatively unlikely (hardware failure causing loss of completed code, Perhaps).

The first stage of in any risk exercise is to identify the risk that might affect the duration or resource costs of the project. A risk is an event that might occur and will, If it does occur, create a problem for the successful completion of the project. In identifying and analyzing risks, we can usefully distinguish between the cause (or risks) and Its immediate effect (i.e.) The problem that it creates and the risk that it will pose to the project.

**Application Factors**

The nature of the application whether it is a simple data processing application, a safety critical system or a large distributed system with real –time elements is likely to be
critical factor. The expected size of the application is also important. The larger the system, the greater is the likelihood of errors and communication and management problems.

**Project Members Factors**

The experience and perfects skills of the team members involved are clearly major factors. Any experienced programmer is one would hope, less likely to make errors than one with less experience. However we must also consider the appropriateness of the experience – experience in coding small data processing modules in Cobol may be of little value if we are developing a complex real-time control system using ETL (Informatica). Such factors as the level of staff satisfaction and the staff turn-over rates are also important to the success of any project – demotivated staff or key personnel leaving unexpectedly have caused many a project to fail.

**Project Factors**

It is important that the project and its objectives are well defined and that they are absolutely clear to all members of the project team and all key stakeholders. Any possibility that this is not the case will pose a risk to the success of the project. Similarly, an agreed and formal quality plan must be in place and adhered to by all participants and any possibility that the quality plan is inadequate or not adhered to will jeopardize the project.

**Project methods**

Using well-specified and structured methods (such as PRINCE 2 and SSADM) for project management and system development will decrease the risk of delivering a system that is unsatisfactory or late. Using such methods for the first time, though, may cause problems and delays – it is only with experience that the benefits accrue.

**Hardware/software factors**

A project that requires new hardware for development is likely to pose a higher risk than one where the software can be developed on existing (and familiar) hardware.
Where a system is developed on one type of hardware or software platform to be used on another there might be additional (and high) risks at installation.

**Changeover factors**

The need for an ‘all-in-one’ changeover to the new system poses particular risks. Incremental or gradual changeover minimizes the risks involved but is not always practical. Parallel running can provide a safety net but might be impossible or too costly.

**Supplier factors**

The extent to which a project relies on external organizations that cannot be directly controlled often influences the project’s success. Delays in for example, the installation of telephone lines or delivery of equipment may be difficult to avoid – particularly if the project is of little consequence to the external supplier.

**Environment factors**

Changes in the environment can affect a project’s success. A significant change in the taxation regulations could, for example, have serious consequences for the development of a payroll application.

**Health and safety factors**

While not generally a major issue for software projects (compared, say, to civil engineering projects), the possible effects of project activities on the health and safety of the participants and the environment should be considered. BS 6079 states that ‘every project should include an audit of these specific risks before work starts’ and that ‘audit updates should be scheduled as part of the overall project plan’.
6.4. Analyzing the impacts of the Factors identified

The Primary objective of this study was to identify the major factors that contribute to chaos in software development. Pareto analysis identified 12 such factors. The role of these factors in contributing chaos in the software development process is analyzed below.

6.4.1 Incomplete Requirements or Project Delivered successfully Situations

The Study revealed that this component contributes most (10.5%) to the chaos in software development. If functional, performance, and reliability requirements are not documented, then each project team member and stakeholder inevitably will have different expectations and assumptions about the project, because each participant is working from a different mental blue print. Asking for sign off on requirements documentations forces differences in expectation assumptions to the surface where they can be resolved. If everyone is not pulling the oars in the same direction, the entropy and therefore the chaos will increase and the project would founder. Projects with undefined success criteria by the definition cannot success. Stake holders who must provide resources and support for the project will not do so, or will soon withdraw those resources, if the objectives and benefits have not been articulated. A Project with undefined success criteria is fated to disappoint. The findings from the study can be summarized as.

Symptoms of chaos:

- Failure of client satisfaction through low level design
- Client stratification failure in project team members due deliverables. (check)
- Clients want often expense of the developers.

Cause of Chaos:

- The main reason is client requirement is not gathered by project team through documents.
- Project delivery will affect the client business. Because of client production oriented projects
- Mainly developer has not satisfied the client with perfect documentation
Solution to this chaos

- Design low level design of particular project submission to the client approval.
- Deliver the project to the client within the mentioned period.
- Developer has to satisfy the client with perfect project documents to sign in delivered agreement.

6.4.2 Documentation Completion failure and late delivery:

During the project delivery consists of so many formal prospects. All of the roles and responsibilities come under project management team. Mainly, Project management is responsible to deliver the documents with project development team. The project management team has to face the deadline. If delivery documents delay the client stratification will be failure. The whole Project development team has to know the problems facing in project.

The team Completion of the project or unable to deliver the project within the particular period. Authorized persons have to face the risks. Hence chaos (8.5%) in the system development. Mostly this cause comes between the project management team and development team. The project delivery is not satisfied the client the findings problems are summarized as.

Symptoms of chaos:

- It is very difficult from the client to receive agreement copy of project delivery.
- Client stratification failure. They need clear documents.

Cause of chaos

- Delivery deadline are uncountable from the client.
- Gathering the requirement was missed from team. Approximately assumed.

Solution to this Chaos:

Project Management team and project development team have well developed plan to deliver the project.
6.4.3 No Proper Project Planning or Project Management Team Failure to handle:

Project Planning means Proper Planning of that particular project Gathering requirement from client. Low level designs, Project Specifications. Allocating Resources to that project very Late. Proper schedule is required in project Management Team [Harold 2013]. To proper Initiation of that particular project, Proper Planning, Identifying stake holders, Risks Management, Time management, Quality Standards, coding standards. Acceptance criteria and so, on. In this Chaos(8%) in the system development projects Project delivery time Team have to check proper Testing. Unit Testing, Block Box Testing, White Box, testing, Integration testing.

The project Plan and schedule is the road map that describes how the project is going too completed. Chaos will result if project Plan is incomplete or not up-to-date. The findings from the study can be summarized as.

Symptoms of chaos:

- The project Plan failure from starting onwards.
- The project went out of control and cannot resolve the problems.

Causes of chaos:

- The project Management not given Proper timing for the particular project schedule.
- Project Management Team is allocated only 20% or 30% timing is allocated for schedule.

Solution to this Chaos:

The project Management Team should have to give Proper schedule. Project Initiation, Project Planning, and Time management [2]. The project Management team should continuously meet the development team members to update the project Status.

6.4.4 Unclear Objectives

The Study revealed that this component is responsible for 7.5% of chaos in software development. Before the start of Project, The project Manager should make sure
that the project objectives are clearly defined and documented along with other requirements. The project manager must communicate this to all the stakeholders and get their agreement. Any differences of opinion need to be resolved before the project starts. Unless the project has a set of documented specific objectives, it cannot be a success. The study revealed that sometimes the project manager feels the pressure to get the project started right away without understanding or defining its objectives. This would increase the chaos in system development and could eventually end up in system failure. The findings from study can be summarized as.

Symptoms of chaos.

- Project team failures in general objectives
- Late running of mappings.

Cause of chaos.

- Specification oriented documents not prepared properly.

Solution to this Chaos:

The project Management team has to make sure Project Specifications and documented or clearly defined with other requirements.

6.4.5 Unrealistic Time Frames.

Unrealistic time frame is responsible for 7% of chaos in software development. Often the customer asks for schedule estimation well before the requirements are determined. However, schedule estimation with acceptable level of accuracy can be obtained only at the end of the requirements determination. Estimating schedule requirements before the project begins can lead to either over-or underestimation of the project schedule.

Symptoms of chaos:

- No, challenge or pressure
- Hasty schedule
- Less Productivity of team members.
Cause of chaos:

- Unrealistic schedule
- Failure to update the schedule with change requests.

Solution to this Chaos:

- Make more realistic initial schedule.
- Keep telling customers how long and how much every change request will cost.

6.4.6 Technology Incompetence

Technology incompetence results in 7% of chaos in software development. Productivity will suffer if the tools being used are not mastered by those using them. It is better to avoid using new technology on projects that have any sort of constraint on time or budget. If the needed skills are not there to start with, then the project management needs to make sure that they are acquired. Projects dealing with high technology need managers with sound technical skills. Skill-driven challengers are not limited to management. Poor developers can sap productivity and make critical and expensive errors.

Symptoms of chaos

- Less Productivity
- Late running
- Team members over scheduled.

Cause of chaos:

- Lack of skilled and experienced manager and/or team members

Solution to this Chaos:

- Make sure that the team managers/members have experience with the chosen technology/methodology.

6.4.7 Communication Breakdown

This Component is responsible for 6% of the people-centered chaos. Any significant project has multiple stake holders and requires an ongoing choreography of various tasks and resources. If all stake holders do not communicate and work together on an ongoing basis, the project team will be pulled in multiple directions. If consensus on project
success criteria is lost, there may be little hope of completing the project on time and on budget, or perhaps of completing it at all. The Study revealed that the lack of communication has a major impact on mountings the chaos within system development. The findings from the study can be summarized as:

Symptoms of chaos:

- Team members do not have the information they need causing delays, issues or changes are not escalated.
- Project reporting is sluggish.

Causes of chaos:

- The project communications plan was not completed.
- The project communication plan does not have enough details.

Solution to this Chaos:

- Find out the communication requirements of all team members and stakeholders,
- Document then in a communication plan, and follow the plan.

6.4.8 No or Insufficient Risk Management Documentation and process:

The Possibility of risks is an important factor towards increasing the entropy and hence the chaos (6%). If risks are not managed timely and effectively, they may even cause failure of the software project. As nothing can be predicated about what will happen in future, the project manager has to take the necessary steps in the present to meet any uncertain situation in the future. Risk management means not only examines and manages the risks inherent in the complexity and size of the proposed system but also includes risks associated with a technical solution to the problem, a well-defined risk management plan should be in place. Unfortunately, in most of the systems selected for the study, there was no clearly defined risk management policy.

The findings from the study can be summarized as:

Symptoms of chaos:
• Unexpected events cause delays.
• Domino effect of things going wrong.

Cause of chaos:
• No formal risk management.
• Just try to predict the big things that can go wrong.
• The sum of all the little things that make a project late.
• Stress of trying to meet schedule when unexpected things keep happening.

Solution to this Chaos:
• List all the works as clearly as possible
• Figure out the what can go wrong with each piece of work
• Prioritize each risk
• Sort the list
• Create a plan to deal with the risks at the top of the list.

6.4.9 Changing Requirements and Specifications

This is Problem is mainly identified in so, many projects as another major source (5.5%) of chaos in system development. The scope creep is more of a problem with traditional methodologies than with modern methodologies like agile. The Volume of change request usually is a barometer of other more significant problems associated with the project. Not knowing exactly what a project is aiming to deliver is a recipe for failure.

The findings from the study can be summarized as:

Symptoms of chaos:
• Schedule and cost over runs
• Incomplete requirements

Cause of chaos:
• Inadequate definition of functional and nonfunctional requirements
• Lack of initial understanding of the business problem being addressed.

Solution of this chaos:
• To ensure that the business case requirements and scope are clearly defined and documented and make sure that the stakeholders understand them and sign them off.
• Stickly rigidly to the scope and if changes are documented, justified and then agreed
• Do not consider the change requests themselves as the problem. Look for the root causes and revise/enhance the parts of the methodology that specifically address the issue.

6.4.10 Project Team Members are over scheduled

This component is responsible for 5% of chaos in the software development Process. Starting a project with insufficient skills or personnel is bound to lead to failure, particularly if the project has schedule constraints. Wrong project resource estimation and lack of a project team with the right skill mix are found to be the root causes of this problem (Leon et al. 2006). Insufficient resources, particularly personal, can also increase the work load on the existing team members. A team working more than 60 hours a week for a sustained period can lead to disastrous situations. The findings from the study can be summarized as:

Symptoms of chaos:
• Less Productivity.
• Developers wasting time in comments.

Solution Of this chaos:
• Make sure that the team members have experienced in the chosen technology / Methodology
• Select Particular skills certified persons for project.

6.4.11 Weak Project manager / Project Leader:

This Component is responsible for 5.5% of the chaos in software development. Effective leadership and communication skills are essential for successful implementation of a software project. Project Managers and project leaders not having leading capacity coming from production area oriented. If they are not having suitable
experience in project management Project get failure. Project manager and project Leader is whole responsible for risk management, time management, project scope (Leon et al. 2006). They are responsible client co-operation, Gather the requirement (Harold 2013). Low level designs, deliverables to the client. The finding from the study can be summarized as:

Symptoms of chaos:
- Communication Failure.
- Lack of co-ordination between the stake holders, user and project Team.

Cause of Chaos:
- Lack of project leader and not maintain quality to deliver the client.
- Lack of communication between client and project Team

Solution of this chaos:
- Project manager/Project leader profile should verify before assign the project.
- Check whether he is skilled person or not.
- Check his previous project. It success or not.

Figure: 6.1 Major Causes of Project Failures
Weak commitment of project Team:

This component is responsible for 4.5% of chaos in software development. No matter how efficient project manager is, he alone cannot carry a project to success. The project manager can successfully carry out his tasks only when he has the support of a committed team of developers (Harold 2013). Project team members with a weak commitment to the project scope and schedule can always find other worthwhile activities to work on. Findings from the study can be summarized as:

Symptoms of chaos:
- Project Team slowly doing the work
- Each and every time deliverables delay to the client.

Cause of Chaos:
- Project Team failure in every work. Which is they are taken.

Solution of this chaos:
- Make sure the schedule
- Project Team members have to follow strictly