

PROJECT CASE

Student Name(s): Averbeck, Michael
Badola, Abhay
More, Rishikesh

Company Name: Grass Roots Software

Division/Business Unit (if applicable): NA

Project Case Title: Understanding Collaboration in a Software Development Environment

Name(s) and Title(s) of individuals included in the project case:
Subodh Gauniyal
Director, Technology Solutions

Sources: Subodh Gauniyal
Director, Technology Solutions
www.grassrootsoftware.com

IT-based Business Climax/Dilemma:

Carr famously said 'IT doesn't matter'¹ and refuted the value of IT as a tool to give any firm strategic advantage. But even back then he saw the patterns in companies' expenditure and could notice a rise in companies' IT expenditure. Some companies even spent 50% of their capital expenditure on IT alone. This remains true to the day; IT department is one of the costliest departments in a company. And this trend doesn't seem to go away.

With expensive IT department coming into the picture, cost reduction measures had to be implemented. Hiring a contractor or a service based firms have become way too commonplace. This new approach brought in problems of its own. Most collaboration based software development projects, involving a stakeholder and a service based firm, bound by a contract, tend to have some communication flaws. Both these firms are not co-located, sometimes they are not even located in the same time zones. It is very difficult to keep each other updated about their progress and be in synchronization. So they try to implement a process that is an amalgam of agile and waterfall models. But we see a lot more problems here too, it usually involves one of the parties attending these sync up sessions at weird timings and naturally not giving their full attention in these meetings. Even if they were collocated, there is no certain hierarchy involved. It is always very confusing who defines roles the service based firm will take. Some other problems like matching the culture of both the firms, aligning skill set etc. arise. All these factors create a communication gap; the resulting product usually just manages to meet the requires agreed upon in the contract but never is the best solution possible, thereby leaving out some scope for development. The ability of a firm to meet the changing requirements but the making strategic decisions during the development process is largely diminished.

Grass Roots Software:

Grass Roots Software is an IT solutions provider, delivering technology and management consulting services. Founded in January 2014 by Subodh Gauniyal and Shannon Hudson Grass Roots Software has grown to win over some of the major clients in the Greater Cincinnati Area. Both the founders worked with CA technologies for more than 15 years, finally leaving the firm to lay the foundation of Grass Roots Software. They put together a team of individuals that are energetic, inherently curious and multidimensional problem solvers.

A range of services designed to meet all the technology needs is offered at Grass Roots Software. Agile approach to problem solving and collaborative nature along with their technology expertise and experience provides their clients an opportunity to grow and prosper. Grass Roots Software is confident about managing the IT needs of a client using their latest innovations in the use of latest Agile Methodologies and Lean Development principles efficiently to minimize wastage and lower the total cost of ownership by engaging with them. Grass Roots Software offer enterprise software services to medium and large organizations in the U.S. Its service offerings focus on Enterprise Applications, Web Applications, and Cloud Enablement. Grass Roots Software takes pride in providing Business Value.

Greater Cincinnati has the highest number of Fortune 500 firms; all of them are growing at an exponential pace, competing with each other. This growth is complemented by a rapidly increasing IT needs too. Capitalizing on this opportunity a large number of IT service provider have come to birth and grown at a swift rate. The only way to survive this thriving and deadly competition is by satisfying your clients and exceeding their expectations.

Over the course of business, Grass Roots Software was noticing a disconnect between the clients' expectations and their finished product. Their products were either 1) just meeting clients need and not going above and beyond their expectations or 2) falling short of their expectations and having some errors or loss of functionality.

Subodh Gauniyal and Shannon Hudson, both ex-employees of CA technologies, collaborated to form "*Grass Roots Software*" in the month of January 2014. Naturally, their first client was CA technologies, having performed well and constantly meeting their demands, Grass Roots started receiving service requests from other firms too. But soon both of them realized that just meeting expectations repeatedly would not be the right strategy to go about in long term, to have a good relationship or get repeated service requests, especially because of the huge competition that they faced. To maintain a good relation, they need to win the hearts of the stakeholders by exceeding their expectations constantly and develop the best possible solutions for them. So in the month of June 2014, Subodh and Shannon turned their focus towards improving their business strategy.

Once they identified the areas for improvement, both of them decided to take action to ensure that these get resolved. This involved taking a hard look internally to see where this disconnection was occurring within their organization. They found that there were multiple points in their processes where slippage in information was suspected. First and foremost, they realized that the requirement gathering stage of the project is not being performed with the best capacity. To design a best possible product, they need to understand the requirement to a much greater detail than just from the documentation provided. The reason that collocated in-house teams perform better than a service provider is because they have a much greater understanding of business and the processes involved; that is how the solution that in-house teams come up with is most aligned. To counter this drawback, in August 2014 they started a unique engagement strategy where they had multiple sessions with the stakeholders and discussed their requirements in an effort to understand the need for the solution. Their motto was "Customer-centric approach begins with engaging with the stakeholders at multiple levels to understand not only the project goals, but also stakeholder's business, infrastructure, and processes allowing us to create a perfectly aligned solution."

Implementation of this was not easy and was not well received. When a stakeholder hires a service based firm they have an attitude that they won't have to think of them or deal with them for a time until the day of release when they receive a perfect solution. It was a difficult task convincing the stakeholders to spend time on helping them understand their business processes. But this all seemed worth when their final deliverable was better than what was expected.

But even after implementing this strategy, Subodh and Shannon realized that there was some more scope for improvement. Having a good business understanding was great, with this business understanding they could set project goals and milestones. But the goals and milestones could be much more aligned if they could be able to involve stakeholders in this process. They could easily misjudge the situation and set up goals that are good but not perfectly aligned with what the stakeholder wants to achieve. Insights from the stakeholders could be much more helpful at this stage. So in the month of March 2015, Subodh and Shannon decided to come up with a collaborative strategy and they started working towards devising it.

This strategy soon came to be looked favorably by stakeholder because it turned out to be time efficient for them as there were no extra development iterations. But there were certain new challenges faced during the actual implementation phase. To get the best possible output from the implementation phase in the small project, the methodology that best fits is agile methodology. That way you keep track of all the changes and updates that are happening around the system and adapt the development strategy accordingly. In this scenario, it is difficult to implement a complete agile methodology because there would be a lot of back and forth of information exchange which is a problem especially when you do not collocate with the stakeholders. Here again, they were facing a very similar problem, the need for the stakeholder to be involved in the development life cycle and thus building up a communication barrier.

In October of 2015, Subodh and Shannon decided to address all three issues by coming up with a consolidated solution. Subodh and Shannon worked part time on this along with two other engineers. They typically put in two hours per day Monday through Thursday, to work and brainstorm towards a strategy that could not just give a consolidated solution but also standardize the methodology for them. They typically had brainstorming sessions and detailed minutes of these meetings were maintained. Thus consolidating their experiences and ideas, they realized that the key question to be addressed were:

- During the sales pitch and initial conversations with the stakeholders.
 - How can they be open and forthcoming with their strengths and weaknesses?
 - How do they demonstrate their culture and the ways they typically overcome certain issues and setbacks?
- During the business and process understanding phase.
 - How do they convince stakeholders to take out time to address this issue?
 - How do they maintain a good communication channel?
- During the requirement gathering stage, and setting goals and milestones.
 - If they should take a deeper look into the company which they were serving and look at the culture?
 - If they should look into the system that made the most sense when looking at how IT is utilized within their company?
 - If they should build a system that the stakeholders want or a system that they NEEDED that would ultimately be the best fit?
- During the development stage.
 - How do they keep open lines of communication with the stakeholders?
 - How you they give updates regarding the system to the stakeholders?
 - How do they demonstrate the functionality of the system before the delivered the final project?
 - How do they ask the stakeholders for any updates from their end to see if the system was still in the state it was during the requirements gathering?

After asking these questions about their processes it started to become clear that communication was a big reason behind the problems they were facing. They needed to change the way they interacted within the company as well as externally with the clients. They decided to take greater care in the front-end work and meetings in order to alleviate some of the stress they were having towards the end of their projects.

After about a month, the team came up with an approach which was eventually called ENCORE: Engage, Collaborate, and Realize.

ENCORE: Engage, Collaborate, and Realize:

When it comes to methodologies, each project is different and each project team is different—there's no one-size-fits-all solution⁵. The software systems are much more complex. From complexity comes the difficulty of communication among team members which leads to product flaws, cost overruns, schedule delays.

In reality, following Agile leads to immutable decisions made by business people with no real understanding of technology. Those decisions are then forced on to developers. The end result is the same as Waterfall, only the names have changed⁶.

The addition of the daily “scrum” ensures that anyone who is perceived to be working too slowly (whether this is true or not) is immediately highlighted. This type of “boiler room” atmosphere may work great for a few, but for most, it is a source of significant stress and ultimately lower productivity. This pressure encourages developers to not think about the future, but rather just slap something together today that just barely functions⁶.

The encore approach is designed to match the existing work culture and the development methodology of the stakeholder firm. It sets up the basis of a collaborative plan that leads to a successful project. It is built upon a very simple but powerful ideology of ENGage, COLlaborate, and REalize. ENCORE approach has given them confidence that they can achieve the best possible strategic solution that will make the stakeholders happy and guarantee repeat business. (Please refer Exhibit 2).

Taking a deeper dive into the ENCORE approach and three principles surrounding it:

Engage:

- Engaging with the stakeholders at multiple levels to understand the project goals.
- Discuss the merits of the feature to be developed, how it could be implemented, etc. These discussions should be vigorous and disruptive (this doesn't mean be unprofessional).
- Ask, is there any already existing module which can be reused?
- Try to get a bird's eye of the project. Look at how the product fits in the clients' environment.
- Engaging with the stakeholders at multiple levels to understand the Business processes and infrastructure involved.
- Engaging with the stakeholders to understand the culture and values of the stakeholder firm.
- Have conformity in the product. Software must conform to the already existing systems or standards.

Collaborate:

- The Domain insights and technical expertise of the Grass Roots software along with business insights of the stakeholders ensure that the project milestones are aligned with the business objectives.
- Collaborated work on this part can guarantee adaptation to any changes that happen during the course of setting goals and milestones.

Realize:

- Implementation phase obviously gets special attention, working together on this phase ensures that the solution is aligned with the business requirements.
- This approach reduces any risk of misaligned as any changes on the stakeholder's side are communicated promptly.
- This approach reduces the time required to develop the solution by avoiding iterations.
- Reduces development time directly translates to a reduced cost of development.

Collaborative Engagement Model:

Having ENCORE approach at its heart, this is a simple yet powerful methodology. This methodology overcomes the traditional weakness related to the lack of communication and lack of collaboration. It is designed to optimize costs of operations and delivers high-quality solutions that are perfectly aligned with the business requirements. Also, adding a tiny bit of adaptability to the culture and processes followed by the stakeholder firm guarantees success. (Please refer Exhibit 3)

The key features of collaborative engagement model that Grass Root software boasts about are:

- Flexible and Collaborative Workflow.
- Minimize Governance and Control.
- Professional with Right Skill Set.
- Trusted Partnership.
- Right Team at the Right Time.

Exhibit 1

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(Original): <http://agilemanifesto.org/>

Exhibit 2

Market/competitor analysis :

SORT: RANK 1 Company								
Rank	Company	2014 Local Sales	Business Name (*Not Previously Ranked)	2014 Local Sales	Local Employees	Local Offices	Services Offered*	Top Local Executive(S)
1	Pomeroy 1020 Petersburg Rd. Hebron, KY 41048 859-586-0600 pomeroy.com	\$642 million	Pomeroy	\$642 million	1,000	1	Managed IT infrastructure services, professional staffing services and procurement and logistics services	Chris Froman, President & CEO
2	CBTS 4600 Montgomery Rd. #400 Cincinnati, OH 45212 513-841-5000 cbts.net	\$384 million	CBTS	\$384 million	840	3	Unified communications, infrastructure as a service, cloud services, data connectivity and network solutions, information security, IT infrastructure managed services, professional services	Scott Seger, SVP/General Manager
3	Sogeti USA 4445 Lake Forest Dr. #550 Cincinnati, OH 45242 513-824-3015 sogeti.com	\$190 million	Sogeti USA	\$190 million	850	3	Project management, architecture, development and testing of BIM, PLM, ERP, open source, CRM and cloud solutions	Bill Blaxton, SVP
4	Roundtower Technologies Inc. 4555 Lake Forest Dr. #220 Cincinnati, OH 45242 513-247-7900 roundtower.com	\$136 million	Roundtower Technologies Inc.	\$136 million	80	1	Data storage, data protection, visualization, networking and security, archive, managed services, professional services	Stephen Power, Owner Stephen West, President
5	Itelligence 10856 Reed Hartman Hwy. Cincinnati, OH 45242 513-956-2000	\$129 million	Itelligence	\$129 million	NA	1	Provides a broad range of consulting and customer support services to further maximize SAP solutions	Steve Niesman, President & CEO

Source(Original): <http://www.bizjournals.com/cincinnati/subscriber-only/2015/03/13/it-consulting-firms.html>

Exhibit 3:

ENCORE



[Services](#) [Products](#) [Work](#) [About](#) [Contact Us](#) 

ENCORE – A Proven Approach to Successful Projects

ENGAGE – COLLABORATE – REALIZE

Our ENCORE approach gels with your existing development methodology, and creates a very effective collaborative layer for successful projects. Driven by our simple, yet powerful philosophies of Engage, Collaborate, and Realize, our ENCORE approach leads to most the strategic solution in every engagement.



Engage

Our customer-centric approach begins with engaging with you at multiple levels to understand not only the project goals, but also your business, infrastructure, and processes allowing us to create a perfectly aligned solution.



Collaborate

With our ENCORE approach, your business knowledge and our technical expertise come together to ensure we arrive at the most strategic solution. We leverage business insights to ensure project milestones are aligned to your business objectives.



Realize

Project execution always gets our undivided attention – after all we're hired to deliver a successful project. Our collaborative ENCORE approach allows us to source projects with the right resources at the right time to ensure a cost-effective, and value-driven execution.



Exhibit 4:

Collaborative Engagement Model



source projects with the right resources at the right
time to ensure a cost-effective, and value-driven
Services Products Work About Contact Us

Collaborative Engagement Model

Optimize cost of operations and deliver high-quality projects every time.

Our core guideline of ENGAGE, COLLABORATE, REALIZE, is at the heart of the Collaborative Engagement Model. It's a simple yet powerful methodology that co-opts the customer into the project workflow and creates a collaboration layer within the engagement.

The collaborative workflow is designed to optimize the cost of operations and deliver high-quality projects.

Our engagement model is designed to overcome traditional process weaknesses by aligning to your business vision and tailoring solutions that create value for your customers.

✓ Flexible and Collaborative Workflow

✓ Minimize Governance and Control

✓ Professional With Right Skill Set

✓ Trusted Partnership

✓ Right Team At the Right Time

Exhibit 5:

Waterfall Software Development Methodology:

Waterfall software development model was formally described by Winston W. Royce in 1970³. It is a sequential development model. The main idea of the waterfall model is that software development proceeds from one phase to the next step in a purely sequential manner. So, the waterfall model means that the programming team should move to a next following step only when the previous phase is completed and everything is approved. However, in some model modifications some slight or major variations and even changes may be included.

The most important advantage of the waterfall idea is that it allows for departmentalization and managerial control. In the common practice, a schedule with deadlines for each phase of project development is made and a product can proceed through the process of developing the created schedule.

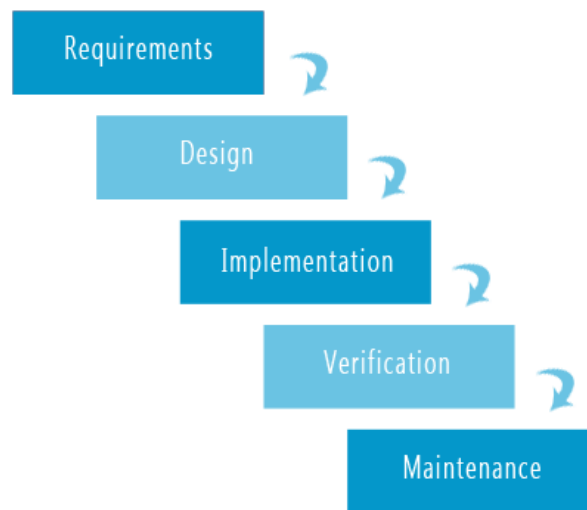
The next argument for the waterfall software development model is that it puts solid requirements on the project documentation (design documents and requirements specs) as well as the source code. And if any of team members left the project less knowledge would be lost in comparison to a project with less designed and documented methodologies.

Waterfall development model implies that a fully working design document should be present so that a new team member or even the entire new team will easily familiarize themselves with a project by reading the documents. In theory, such process leads to the project being delivered on time because of a detailed plan for each phase created previously.

However, in practice, it's often impossible to follow the pure waterfall model because it does not consider the inevitable changes and revisions that become necessary in the development process. Also, many specialists argue on the waterfall model as a bad idea in practice, generally because of the inability to perfect one stage of a software product, before moving to another stage and learning something new from them for any non-trivial project.

For example, customers may not be sure of what requirements exactly they want before they see a working prototype and can comment upon it; they may change their requirements constantly after seeing a result of development, and program designers and implementers may have little or no control over this. If customers change their requirements after a design document is finished, then the document must be modified to correspond to the new requirements. Also, project designers may not be aware of future implementation or development difficulties when writing a design for an unimplemented software product. For example, it may become clear in the implementation phase that a particular area of product functionality is extraordinarily difficult to implement.

In this situation, it is better to revise and change the design document rather persist in using a conception, that was made based on incorrect predictions and that does not consider the newly discovered problem areas. This is one of the reasons, why the waterfall model isn't commonly used in agile software development.



Source: <http://www.oxagile.com/company/blog/the-waterfall-model/>

Exhibit 6:

Agile Software Development Methodology⁴:

Agile isn't a set of practices but a mindset for approaching software development. Agile takes into account that the team doesn't have all the answers and doesn't subscribe to the silver-bullet theory⁵.

It recognizes the importance of process and tools, with the additional recognition that the interaction of skilled individuals is of even greater importance. Similarly, comprehensive documentation is not necessarily bad, but the primary focus must remain on the final product—delivering working software. Therefore, every project team needs to determine for itself what documentation is absolutely essential. It iterates that only through ongoing collaboration can a development team hope to understand and deliver what the client wants.

- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.
- Business people and developers work together daily throughout the project.
- The most efficient and effective method of conveying information with and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

References:

1. Carr, N.G. (2003, May) "*IT doesn't matter*" [Harvard Business Review] <https://hbr.org/2003/05/it-doesnt-matter>
2. Christensen, C., Hall T., Dillon K., Duncan D., (2016, September) *Know you Customers' "Jobs to be done"* [Harvard Business Review] <https://hbr.org/2016/09/know-your-customers-jobs-to-be-done>
3. W. W. Royce, "*Managing the development of large software systems: concepts and techniques*": <http://dl.acm.org/citation.cfm?id=41801>
4. The Agile Manifesto: <http://www.drdoobs.com/open-source/the-agile-manifesto/184414755>
5. Frederick P. Brooks, Jr. "*No Silver Bullet*": <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1663532>
6. Agile Is the New Waterfall: <https://medium.com/swlh/agile-is-the-new-waterfall-f7baef5d026d#.zfdteylq>
7. Grass Roots Software, about page: <http://www.grassrootsoftware.com/about/>