Customer Relationship Management (CRM) & best practices in CRM

written by Manoj Khanna | April 13, 2003

CRM Implementation

Customer Relationship Management solutions are often a multi-million dollar investment. After committing to such a large expenditure, companies must be assured that their CRM investments will thrive. Recent research shows that a customer-focused culture is just as critical as the technology to a CRM implementation success. Some focused points:

Most useful performance metrics and how performance measurement can be used to build support :

1. Ramping up CRM from the pilot stage to global roll-out
2. Ensuring that ownership and support for the initiative are company-wide
3. The knowledge to transform a corporate culture from product-focused to customer-focused allows leading companies to attract, nurture and retain highly profitable customer relationships

Study Snapshot

Globalization has widened customer choices by expanding a purchaser’s reach across countries and continents. E-business tools have made it easier for consumer and business customers to evaluate diverse purchase options. Consequently, the job of building lasting customer relationships has grown far more complex for most companies. Many companies now have corporate-wide contact and account management systems. However, the strategic and cultural challenges of CRM continue to hamper

successful implementation that produces positive financial results. In short, CRM implementation challenges often stand in the way of companies realizing the full economic benefits of customer-focused growth strategies.

CRM initiatives must be carefully planned and consistently supported in order to drive customer retention, cost reduction and higher profitability. Many companies proclaim that customer relationships are critical to lasting marketplace success. Yet few companies have successfully shaped their marketing, sales and delivery systems to optimize return on customer relationship investments. This benchmarking study is designed to serve as a roadmap to this end.

Milestones in best CRM Practices

The various studies on CRM Implementations, details the key milestones and practices we have used to successfully execute CRM initiatives. Some milestones:

4. Integrating Legacy Initiatives
5. Allocating Resources for Pilot Programs
6. Distinguishing Global and Local CRM Processes
7. Leveraging Knowledge Management Tools
8. Developing Customer-Centric Employees

These practices outline how best-in-class companies turned their multi-million dollar CRM investment into a multi-faceted approach to retaining and maintaining valuable customer relationships through the use of effective CRM Practices.

© Manoj Khanna 2003 – 2012.

Customer Relationship Management (CRM) & best practices in CRM

written by Manoj Khanna | April 13, 2003

CRM Implementation

Customer Relationship Management solutions are often a multi-million dollar investment. After committing to such a large expenditure, companies must be assured that their CRM investments will thrive. Recent research shows that a customer-focused culture is just as critical as the technology to a CRM implementation success. Some focused points:

Most useful performance metrics and how performance measurement can be used to build support :

9. Ramping up CRM from the pilot stage to global roll-out
10. Ensuring that ownership and support for the initiative are company-wide
11. The knowledge to transform a corporate culture from product-focused to customer-focused allows leading companies to attract, nurture and retain highly profitable customer relationships

Study Snapshot

Globalization has widened customer choices by expanding a purchaser’s reach across countries and continents. E-business tools have made it easier for consumer and business customers to evaluate diverse purchase options. Consequently, the job of building lasting customer relationships has grown far more complex for most companies. Many companies now have corporate-wide contact and account management systems. However, the strategic and cultural challenges of CRM continue to hamper successful implementation that produces positive financial

results. In short, CRM implementation challenges often stand in the way of companies realizing the full economic benefits of customer-focused growth strategies.

CRM initiatives must be carefully planned and consistently supported in order to drive customer retention, cost reduction and higher profitability. Many companies proclaim that customer relationships are critical to lasting marketplace success. Yet few companies have successfully shaped their marketing, sales and delivery systems to optimize return on customer relationship investments. This benchmarking study is designed to serve as a roadmap to this end.

Milestones in best CRM Practices

The various studies on CRM Implementations, details the key milestones and practices we have used to successfully execute CRM initiatives. Some milestones:

12. Integrating Legacy Initiatives
13. Allocating Resources for Pilot Programs
14. Distinguishing Global and Local CRM Processes
15. Leveraging Knowledge Management Tools
16. Developing Customer-Centric Employees

These practices outline how best-in-class companies turned their multi-million dollar CRM investment into a multi-faceted approach to retaining and maintaining valuable customer relationships through the use of effective CRM Practices.
© Manoj Khanna/Open Source World/rapidblog.com 2003, 2004, 2005, 2006, 2007, 2008, 2009
Powered by Dextrus Prosoft, Inc.

Developer Vs. PM's perspective

written by Manoj Khanna | April 13, 2003

Estimation & Invention

We simply can't estimate by the minute. Some teams estimate by days and is more often wrong then right. There are simply too many facts not controllable by a project manager to give an estimate by minute a meaning.

We often say: We INVENT things. Every program deals with a new thing and it's nearly impossible to estimate the needed time for inventions.

As far as I can see PM and the team should use something to track down the time actually needed and compare it with their estimates. This will give them a better idea on how and where their estimates were wrong. Over time their estimates will improve by doing this.

Bug Fixing

Estimates for fixing a bug are one of the hardest things to do. This is because very often you need some idea where to start checking things. Even if you got this idea, it is often not clear what exactly needs to be changed. I find myself often in a situation, where some modules simply don't work together very well. It would be easy to change one of the components to fit the needs, but then a 3rd module would break, so as one possible option I have to accept a hack in the 2nd module. Or I can rearrange the whole thing of 3 modules to work together perfectly (preferred way!) These decisions can only be made when the clock for fixing the bug has long started ticking.

KNOW where to SAVE

Some people think that software engineering is an expensive work. And often the only thing that counts is the time used to get a program running. The quicker, the better. Regardless of the remaining errors, the maintainability of the code or its documentation.

I'd say don't accept this. This is short time thinking. If you need two days more to get it done correctly, do it now. The "saved" two days WILL come back later (maybe a year or two years later), but they WILL come back, but then it will be a week.

Coders Haven…where??

One of the best methods the coders follow is "Think first, code later". The theory behind this is that the time spent in design pays back by lowering coding time. As it is widely known that:

"How do you know what to code, if you have no idea of the problem?". And Moreover structured programming is the only way to get larger projects done.

There is a saying:

"Plan to throw one away! You will anyway".

Fact is: If you analyzed the problem and thought about different solutions, about their pitfalls, drawbacks and advantages you will end with a design that fits your needs. But sometimes it happens that it turns out that your theoretical concepts don't work that good in practice. Even if it is hard: Throw it away!

I will quote here an example of my previous company where I was reworking on the code that one of our former employees had left. He was one of the guys, who fixed a bug by adding a special layer on top of his code to handle a (what he thinks

is a) special case. Needles to say, that this additional code has bugs by itself, so a next layer is needed to take this into account. When I rewrote his code I manage to get rid of 2/3 of his code, making the whole thing smaller and thus more manageable. The whole process took

4 weeks of my time and which I could have finished in one and a half weeks.

Conclusion

Well, if someone plans to make things work without even knowing how, it’s his choice but the last thing to take into consideration is that his program is going to work. Needless to say that the most famous scientists just say: “If you want to make something, build the model first”. Software engineering is just building models for OO and similar programming techniques. If you don’t intend to program using certain methodologies and style you should consider taking track of all the variables through the code. PM is not a coder, but someone who must know what and when to include in the project. After making a good project representing using the best terminologies (estimation and planning) then rest is just a piece of cake.

© Manoj Khanna 2003 – 2012.

Developer Vs. PM’s

perspective

written by Manoj Khanna | April 13, 2003

Estimation & Invention

We simply can’t estimate by the minute. Some teams estimate by days and is more often wrong then right. There are simply too many facts not controllable by a project manager to give an estimate by minute a meaning.

We often say: We INVENT things. Every program deals with a new thing and it’s nearly impossible to estimate the needed time for inventions.

As far as I can see PM and the team should use something to track down the time actually needed and compare it with their estimates. This will give them a better idea on how and where their estimates were wrong. Over time their estimates will improve by doing this.

Bug Fixing

Estimates for fixing a bug are one of the hardest things to do. This is because very often you need some idea where to start checking things. Even if you got this idea, it is often not clear what exactly needs to be changed. I find myself often in a situation, where some modules simply don’t work together very well. It would be easy to change one of the components to fit the needs, but then a 3rd module would break, so as one possible option I have to accept a hack in the 2nd module. Or I can rearrange the whole thing of 3 modules to work together perfectly (preferred way!) These decisions can only be made when the clock for fixing the bug has long started ticking.

KNOW where to SAVE

Some people think that software engineering is an expensive work. And often the only thing that counts is the time used to get a program running. The quicker, the better. Regardless of

the remaining errors, the maintainability of the code or its documentation.

I’d say don’t accept this. This is short time thinking. If you need two days more to get it done correctly, do it now. The “saved” two days WILL come back later (maybe a year or two years later), but they WILL come back, but then it will be a week.

Coders Haven…where??

One of the best methods the coders follow is “Think first, code later”. The theory behind this is that the time spent in design pays back by lowering coding time. As it is widely known that:

“How do you know what to code, if you have no idea of the problem?”. And Moreover structured programming is the only way to get larger projects done.

There is a saying:

“Plan to throw one away! You will anyway”.

Fact is: If you analyzed the problem and thought about different solutions, about their pitfalls, drawbacks and advantages you will end with a design that fits your needs. But sometimes it happens that it turns out that your theoretical concepts don’t work that good in practice. Even if it is hard: Throw it away!

I will quote here an example of my previous company where I was reworking on the code that one of our former employees had left. He was one of the guys, who fixed a bug by adding a special layer on top of his code to handle a (what he thinks is a) special case. Needles to say, that this additional code has bugs by itself, so a next layer is needed to take this into account. When I rewrote his code I manage to get rid of 2/3 of his code, making the whole thing smaller and thus more manageable. The whole process took

4 weeks of my time and which I could have finished in one and a half weeks.

Conclusion

Well, if someone plans to make things work without even knowing how, it's his choice but the last thing to take into consideration is that his program is going to work. Needless to say that the most famous scientists just say: "If you want to make something, build the model first". Software engineering is just building models for OO and similar programming techniques. If you don't intend to program using certain methodologies and style you should consider taking track of all the variables through the code. PM is not a coder, but someone who must know what and when to include in the project. After making a good project representing using the best terminologies (estimation and planning) then rest is just a piece of cake.

© Manoj Khanna/Open Source World/rapidblog.com 2003, 2004, 2005, 2006, 2007, 2008, 2009

Powered by Dextrus Prosoft, Inc.

Open Source Software, GPL & Beyond

written by Manoj Khanna | April 13, 2003

When I heard about open source software back in my college days, it was an all exciting news for me. Open source softwares have been a delight to the developing community and now companies like IBM, and Sun Microsystems are not keeping away from it either. Many large and small corporations feel the advantage of an open source environment and building open source projects and providing solutions thereof.

Quite intrestingly things on the other hand has started to take some other shape, in order to save the common and general developers interests something called GPL has been into

practice for quite some time now to save the developing community efforts. A recent article about Why Open Source Stifles Innovation suggests the threats to the commercial software products in the long run. Needless to say this might have been the innovation of people who developed doors and windows alike. To shut down your car you won't plug on START button!

On the other hand Considering Open Source Software is coming out to be pretty powerful and has given the industry some very intresting features and have tried to kill the monopoly-threat which was posing a couple of years ago. Needless to say that we all live in a democratic world. So the industry should also have a democratic point of view in the development of softwares and the new and innovative products which are of-course for the betterment of the human community at large and its practices.

A small world is a better world and a peaceful world, but when it gets bigger – the big fishes often think themselves as kings and queens – rather working together with the smaller fishes. And when the time comes – the kings and the queens do get their lessons. Working together and as a team is the **MANTRA**. Its not me or I its just **"we"**.

© Manoj Khanna 2003 – 2012.

More Agility in Offshore Outsourcing

written by Manoj Khanna | April 13, 2003

We are moving forward everyday. And one of the biggest factor in this economy today is to save on cost from all possible angles and come out as a winner. Despite some major bans and

threats to offshore outsourcing business, things are moving forward. And they are moving forward in a better way for the ailing economy to forsee profits and bring in some hope. Any busines today wants to save on its expenses. Money saved is money earned. No one will spend $100 when they can get the same work and quality off $20 or even less at times. The difference between here and there is not even a mile away when it comes to communication and exchange of information keeping the physical distance apart.

The idea today is not to hurt ourselves too much and see what's coming. The early runners of outsourcing companies like GE, and Microsoft, TI today are reaping profits and have been doing so in those stiff times too. A recent news on Offshore Street Contracts is not hair raising, but infact a reality from which most of the companies are shying. To accept the challenge is not a beating thing. But only an eye opener of what's coming and standing in front of them.

The future can be predicted. Things will happen and in order to save and survive, outsourcing is for the rescue. So, its upto the companies now whether they want to be early adopters or late adopters.

© Manoj Khanna 2003 – 2012.

Open Source Software, GPL & Beyond

written by Manoj Khanna | April 13, 2003

When I heard about open source software back in my college days, it was an all exciting news for me. Open source softwares have been a delight to the developing community and now companies like IBM, and Sun Microsystems are not keeping

away from it either. Many large and small corporations feel the advantage of an open source environment and building open source projects and providing solutions thereof.

Quite intrestingly things on the other hand has started to take some other shape, in order to save the common and general developers interests something called GPL has been into practice for quite some time now to save the developing community efforts. A recent article about Why Open Source Stifles Innovation suggests the threats to the commercial software products in the long run. Needless to say this might have been the innovation of people who developed doors and windows alike. To shut down your car you won't plug on START button!

On the other hand Considering Open Source Software is coming out to be pretty powerful and has given the industry some very intresting features and have tried to kill the monopoly-threat which was posing a couple of years ago. Needless to say that we all live in a democratic world. So the industry should also have a democratic point of view in the development of softwares and the new and innovative products which are of-course for the betterment of the human community at large and its practices.

A small world is a better world and a peaceful world, but when it gets bigger – the big fishes often think themselves as kings and queens – rather working together with the smaller fishes. And when the time comes – the kings and the queens do get their lessons. Working together and as a team is the **MANTRA**. Its not me or I its just **"we"**.

© Manoj Khanna/Open Source World/rapidblog.com 2003, 2004, 2005, 2006, 2007, 2008, 2009

Powered by Dextrus Prosoft, Inc.

More Agility in Offshore Outsourcing

written by Manoj Khanna | April 13, 2003

We are moving forward everyday. And one of the biggest factor in this economy today is to save on cost from all possible angles and come out as a winner. Despite some major bans and threats to offshore outsourcing business, things are moving forward. And they are moving forward in a better way for the ailing economy to forsee profits and bring in some hope. Any busines today wants to save on its expenses. Money saved is money earned. No one will spend $100 when they can get the same work and quality off $20 or even less at times. The difference between here and there is not even a mile away when it comes to communication and exchange of information keeping the physical distance apart.

The idea today is not to hurt ourselves too much and see what's coming. The early runners of outsourcing companies like GE, and Microsoft, TI today are reaping profits and have been doing so in those stiff times too. A recent news on Offshore Street Contracts is not hair raising, but infact a reality from which most of the companies are shying. To accept the challenge is not a beating thing. But only an eye opener of what's coming and standing in front of them.

The future can be predicted. Things will happen and in order to save and survive, outsourcing is for the rescue. So, its upto the companies now whether they want to be early adopters or late adopters.

© Manoj Khanna/Open Source World/rapidblog.com 2003, 2004, 2005, 2006, 2007, 2008, 2009

Powered by Dextrus Prosoft, Inc.

Agile Project Management

written by Manoj Khanna | April 13, 2003

It was a delightful experience to hear Scott Ambler speak about Agile Methods and Techniques at the Xerox PARC in the heart of Sillicon Valley. And surely he popped different ideas of “the” traditional approach towards a more “agile” approach in management of the project. Does it means we do it fast and try to get rid of it? ASAP. Well surely there is still a long way to go. There is still a long debate about what goes in managing the project so perfectly.

Scott’s whole perspective came from an “agile” developer. But the project is not run by developers. Although developers play an important role in development of the project. The agility in the project management should come from the project manager. And it should come in the early stages of the project. The certain approaches of maintaining the starting the actual product development over documentation is another big debate. But what makes the project management agile?

The agility of the project depends on the entire team, right from the project manager, to the team and to various stakeholders. Although stakeholders are only interested in the end results, but it is their responsibility as well how well they want the project to go. The early adopters of the agile methods produce dramatic results of success with the project. Some of the agile methods already in practice include Daily Scrum and Open Source and Agility.

The conclusion here could not be simpler, but yah this debate is on, and will be on. But the basics are that the project management should balance the principles of ‘agility’ with the business imperatives.

© Manoj Khanna 2003 – 2012.

Agile Project Management

written by Manoj Khanna | April 13, 2003

It was a delightful experience to hear Scott Ambler speak about Agile Methods and Techniques at the Xerox PARC in the heart of Sillicon Valley. And surely he popped different ideas of “the” traditional approach towards a more “agile” approach in management of the project. Does it means we do it fast and try to get rid of it? ASAP. Well surely there is still a long way to go. There is still a long debate about what goes in managing the project so perfectly.

Scott’s whole perspective came from an “agile” developer. But the project is not run by developers. Although developers play an important role in development of the project. The agility in the project management should come from the project manager. And it should come in the early stages of the project. The certain approaches of maintaining the starting the actual product development over documentation is another big debate. But what makes the project management agile?

The agility of the project depends on the entire team, right from the project manager, to the team and to various stakeholders. Although stakeholders are only interested in the end results, but it is their responsibility as well how well they want the project to go. The early adopters of the agile methods produce dramatic results of success with the project. Some of the agile methods already in practice include Daily Scrum and Open Source and Agility.

The conclusion here could not be simpler, but yah this debate is on, and will be on. But the basics are that the project management should balance the principles of ‘agility’ with the business imperatives.

© Manoj Khanna/Open Source World/rapidblog.com 2003, 2004, 2005, 2006, 2007, 2008, 2009

Powered by Dextrus Prosoft, Inc.