This paper focuses on reducing the cost of software to customers and talks in detail about ‘Improving test productivity’ as one of the best practice.

Test teams are often asked, “How do you plan to improve test productivity?” AND there is an opportunity in this question for every test organization within MSIT!!!

Well, there is no one answer for it!!!

On an average overall test activities effort is around 50% of development effort, which can definitely be improved. Test teams needs to start moving towards more tailored test processes which suit their business model. A key challenge for any organization is always how to reduce COST for customer. Today the customers can be made happier
using cost reduction rather than giving ornamental features. **COST, Timelines and Quality are three most important priorities for business in today’s highly competitive market.** If we don’t reduce cost for customers, they will do it in their own way!!! As the IT industry faces challenges due to economical slowdown, we need to find different ways to reduce cost for our customer so that they still see smooth ROI, and remain our customer’s.

This paper talks about causes which refrain test teams from being more productive and suggests measures to overtake them. Use of Automation right from test case preparation till test reporting along with examples/tools available at MS is explained in later sections.

Testers does spend most of their time in test case preparation, execution, and bug reporting and finally test status reporting. This paper will talk about where we lose opportunity to improve productivity at every step. Also, suggesting some productivity tools to do our job better and in more efficient manner. A suggested test lifecycle is explained which is a tailored version of typical Iterative SDLC model, which can help us.

**Problem Definition**

Cost incurred by business in IT is huge is these days, to support a business process. In a typical scenario, IT leadership team suggests adoption/creation of software which will help customer’s operate their business process in a better manner and save costs. If we stop at this step Then over a period of time customers will not see much value. The problem here is their gains from spending on IT are going stable over a period of time but the cost with every release is going higher. If we plot a graph like below, most of the cases business will be asking how to reduce cost! This is something the IT teams can work proactively on by using some best practices.
In most of the projects at MSIT, test team involvement starts with understanding the Business Requirement Documents (BRD) and/or the Functional Specifications, and ends with certifying build which are ready to go through User Acceptance Testing. They do this every quarter or even monthly/bi-weekly basis. Aren’t we following big processes for activities which are repeated in small timelines? It leads to wastage of opportunity to increase productivity. Productivity necessarily does not mean delivering more to a customer NOR does Quality mean following lot of processes!!!

Some of the challenges commonly faced by the test teams which take us away from improving productivity in testing (ultimately overall project) are:-

1- Lack of proper BRD’s 
2- Low quality of Functional Specs 
3- Forgetting the past - Not using learning’s from previous projects 
4- Starting test case creation before Functional Specs are signed-off 
5- Lack of Discipline 
6- Ineffective effort estimations (re-estimations atleast 2-3 times) 
7- Manual test case generation 
8- Little or no use of tools in different phase of testing 
9- Root cause analysis of high efforts not done
The bigger problem is reducing cost to our customers and if we break it up into factors, we can see that test productivity is one of them. Now to understand where we can improve test productivity to reduce our overall problem, we need to understand where we are spending more effort.

Below is breakup of test efforts on various test activities, once we understand which item takes more time, we can go into root causes using below graph. These figures are based on my experience and some research done over WWW. The graph below contains only more frequently and standard tasks, there are other tasks like verification of tasks etc. but for simplicity I am clubbing most of tasks into below items.

![Test Effort Distribution](image)

Above graph clearly indicates that the biggest areas of concern are around test case preparation and execution efforts. Now our problem statements becomes:

**How to reduce cost to customer by improving test productivity and cutting down hours spent on test case preparation and / or execution?**
One thing which we can control as engineering team is productivity. So, let us focus on this going forward in this paper.

I propose following steps to overcome this challenge and foresee respective benefit(s).
1. **Use of productivity tools**

   We need to use more of productive tools like below to cut hours on non-engineering tasks we do:-


   **StickySorter** - [http://codebox/stickysorter](http://codebox/stickysorter)

   **Infinity (Canvas for OneNote)** - [http://codebox/infinity](http://codebox/infinity)

   **Plex for PPT** - [http://codebox/PlexForPPT](http://codebox/PlexForPPT)

   These are the some tools which can help us to do our job better!

2. We need to choose a **tailored test process** when a project is taken up; often this step is missing when a project is in early days and left to days when actual work starts.

   **Benefit:** - We all don’t drive same car!

   We all don’t like same ingredients in our curry!

   We all are different and every project has some difference in terms of the way it should be executed. So we need to choose appropriate steps from any test process defined at organization level, keeping in mind that the quality of the project is at stake but the need is to drive productivity and less process barriers to get simpler things done. *Suggested approach is explained in later sections of the paper.*

3. **Test team needs to get into requirement analysis** along with Analysts and Program Managers. This not only uncovers understanding gaps but helps customers with experience of tester’s and this way Test can better understand the needs and wants of the customer. In some cases test team have suggested features in applications which add great value to overall customer satisfaction!

   **Benefit:** - This can add lot of value to overall understanding of the test engineers in terms of business and allows them to get close to business. This would be a starting point for test team to get into a mode wherein they can add value not only as gatekeeper’s of quality but by suggesting some functionality which can help customers with some smart decisions to save dollar!
4. **Test effort estimation should be driven towards productivity** not only effort to complete a task. Do not plan too much buffer instead plan for productivity.

**Benefit:** - Many times test planning team ends up giving efforts on higher/lower/medium side, but the planning of test effort should be driven towards productive test case efficiency; this will help us in reducing efforts in test tasks and eventually improve productivity. We might spend more time in automation initially but ROI in long run drives great savings.
5. Test team needs to use their knowledge from previous projects to adopt automation /tools/ processes/ utilities which can help them **reduce test case preparation effort**. Manual test case writing consumes lot of time and is a repetitive process, and calls for exploring the existing tools or develop small frameworks to get automated test cases formulated.

**Benefit:** - If test team can cut their test case preparation time by at least 20%, using tools to obtain automatically generated test cases, they save tester from boredom and drives innovation as well as productivity.

6. **Automate Regression/Build verification tests.**

**Benefit:** - Every time application is modified there are areas you need to regress, manually doing it takes a lot of time, we need to drive aggressively towards automation at every possible set of test cases, and this can reduce test case execution efforts by at least 25%

Use more of available tools like RPF, VSTS and tools available at [http://toolbox](http://toolbox), we need not reinvent wheel every time for automation needs, there is lot of work already done and we need to adopt. Test team can use simple stored procedures, scripts to do BVT’s and once BVT is done, testers will also have an idea which functionality looks more suspect-able to bugs. One of the key to successful regression testing is targeting key business workflows.

Tools which can do data quality checks /functional testing need to be emphasized and made part of the test activities, to save big chunk of test execution effort in every cycle.
7. **Automate environment preparation and build deployment**

**Benefit:** - Every time a build/drop comes to test team they spent time in preparation of environment and deploying it, on an average they do this at least 15-20 times in a year. Use scripting languages and deployment tools to save this manual effort and drive productivity.

There is a tool developed by test folks in BI-COE, MSIT which can help us save time on environmental preparation. You can see details at [http://toolbox/EPA](http://toolbox/EPA)

8. **Be more productive in bug lifecycle**

**Benefit:** - Tester needs to start logging bug(s) with all required details and any info which can help Development team fix this faster, this will save turnaround time in the bug lifecycle and bring down effort by not only test team but entire project teams.

Adoption of single tool across all teams like VSTF also helps driving productivity as teams become used to it, and start using all relevant fields faster!

9. **Reduce documentation time**

**Benefit:** - We need to go towards tailored process wherein we do just ample amount of documentation. Many times non-engineering task hours go beyond engineering task hours.

Instead of preparing lot of test documents, just do needful and use existing techniques like traceability matrix, use more of six sigma tools like pareto chart, Fishbone diagram, FMEA etc.. We need to prepare Smart documents rather than big documents!

10. **Automation of test status reporting** needs to be explored/implemented

**Benefit:** - Test team can save lot of time if they automate test status reporting, some manual processes like generation of graphs related to test metrics, collecting data from different configuration tools etc needs to be streamlined. Use of one tool across MSIT for test case/bugs management will help drive automation here.

Use more of VSTF features, which can serve as end-2-end test case management, Configuration management and reporting tool. Features like SSRS reports, TFS Mail subscription can reduce time tremendous here. Sample TFS mail sent for test reporting, which saves tester at least 1 hours REPITATIVE!!!
11. **Proactive design needed in automation suites**

**Benefit:** Most testers refrain from automation because of overhead of maintaining automation scripts, yes, sometimes it can go high, but we need to think of this factor/accept it during test planning phase, if test cases/automation scripts are designed in such a way that they can be changed later then effort involved in maintenance/rework will be less, as you are already planning for it!!!

Key theme I am put across here is Have proactive design rather than reactive design.

The below graph shows we need to emphasis more to reduce cost when it comes to designing automation scripts/suites.

Identify and Track vital failures right from start

![Image of a graph showing cost due to rework in automation across different phases like Planning, Requirements, Design, Implementation, and Testing.

Key points from the graph:
- Most teams focus on Testing (50%)
- Significant cost savings can be achieved by focusing on earlier phases like Planning (10%), Requirements (20%), Design (30%), Implementation (40%)

12. **Any test process which blocks delivery should be revisited**

**Benefit:** Processes’ are to help deliver project in better way, but by no means it mandates you can’t deliver a project which does not follow standard process, as said before in this paper COST, Timelines and Quality are three most critical factors in same order. The value additions processes add to a project are numerous but at any point of time if it blocks
product delivery should be revisited; test team/leads should be open to discussion in such cases.

13. **Improve and track Test Coverage**

**Benefit:** - Test coverage should be given high importance to ensure that we don’t write extra test cases, this will save effort in test case preparation at the same time ensure all functionalities have been covered.
Proposed Approach on Test Life Cycle

Key Features of this proposed approach are:

Test team helps Analysis/PM team in Functional spec (planning phase), the ownership still lies with PM's.

High level Test strategy is identified before actual test work starts.

Think of automation as part of test cycle than optional.

Regression testing only executed as per test exit criteria mentioned in test plan.
Conclusion

Productivity increase in testing can lead to overall improvement in the project. With tried suggestions, we can improve productivity in testing over a period of time, which will eventually help our customer’s a lot & make them happier!

The test process suggested with few tweaks from normal test cycle can again be modified to suit particular needs of the project. Overall mantra as emphasized in this paper is to automate steps which are repetitive and consuming bulk of testing time.
Appendix A

IT – Information technology

MSIT – Microsoft IT

Pareto chart -A Pareto chart is a special type of bar chart where *the values being plotted are arranged in descending order*. The graph is accompanied by a line graph which shows the cumulative totals of each category, left to right. The chart is named after Vilfredo Pareto, and its use in quality assurance was popularized by Joseph M. Juran and Kaoru Ishikawa.

Fishbone diagram - The Ishikawa diagram (or fishbone diagram or also cause-and-effect diagram) are diagrams, that shows the causes of a certain event. A common use of the Ishikawa diagram is in product design, to identify potential factors causing an overall effect

FMEA - A failure modes and effects analysis (FMEA) is a procedure for analysis of potential failure modes within a system for classification by severity or determination of the effect of failures on the system. It is widely used in manufacturing industries in various phases of the product life cycle and is now increasingly finding use in the service industry. Failure causes are any errors or defects in process, design, or item, especially those that affect the customer, and can be potential or actual. Effects analysis refers to studying the consequences of those failures.

ROI – Return on investment

VSTF - Visual Studio Team System 2008 Team Suite provides multi-disciplined team members with an integrated set of tools for architecture, design, development, database development and testing of applications. Team members can continuously collaborate and utilize a complete set of tools and guidance at every step of the application life cycle

Reference links:-

http://library.dzone.com/whitepapers/improve-test-productivity-and-

http://adsabs.harvard.edu/abs/1983flte.confR....W

http://whitepapers.businessweek.com/detail/RES/1157126441_158.html