

# STC 2011: Leveraging Microblogging for Effective Team Collaboration

---

Raj Kamal, [rajkamal@microsoft.com](mailto:rajkamal@microsoft.com), Microsoft, Hyderabad

Gunjan Jain, [gunjain@microsoft.com](mailto:gunjain@microsoft.com), Microsoft, Hyderabad

*Proposing the usage of Microblogging as a project communication layer on top of defect and requirement management tools like QC, primarily for Agile projects, having teams spread across geographies. Also to better help teams to include customer, stakeholders & management in day-to-day project conversations and updates.*

*At one hand we are witnessing the increasing popularity and adoption of different flavors of agile along with inescapable need to have execution of projects from different geographies and time zones and on the other hand, we are witnessing the rise of the realm of Social Media/Microblogging tools like Facebook, Twitter (and why not mention your internal Microblogging tools for ex. built using SharePoint). This doesn't take lot of thinking to realize that two will cross their paths sooner or later. I strongly believe that Social networking/Microblogging can make more profound impact at a smaller grain i.e. at project or team level in an enterprise.*

*I want to quote a text from an internal publication that got review comments from Bill Gates itself. This is just an evidence of how outlook is still used in projects as a social connection tool.*

*"At Microsoft, Outlook is truly an application that 'runs in every office, on every desktop'. This unchallenged domination has obvious and numerous benefits, but there's also a price we pay. Since e-mail by its nature is an application that connects people, over time people at Microsoft got used to using it for the majority—if not all—of their social connection tasks. As of today e-mail is the de-facto social networking application within Microsoft, with Outlook being the windows to the vast majority of our social interactions."*

*With people getting more and more hooked to Microblogging sites, it's becoming quite obvious that Social media is not just a preferred channel for connecting with people for well not just for non-official communication but can also be instrumental in an enterprise environment at a project/team level. At project level it can be quite effectively used by agile teams as a loosely coupled thin-layer on top of our tools*

*like QC for better collaboration and sharing of ever changing information as part of software development processes such as defect management, change management, requirement management to name a few.*

*Yes, we are proposing marriage of QC with your enterprise Microblogging platform. This integration is expected to be loosely coupled but highly cohesive and not the other way around. The idea is not to increase redundancy and process overhead by adding one more layer but instead to make our development process much more effective and agile by providing a richer experience that fosters seamless integration.*

## **Evolution of Communication in Software Development Arena**

This section is presented at a relatively high level without going in deep as the overall objective is to give an overview of changing trend over time to meet the needs.

### **PAST: No integration between Email and QC**

Those days there was no or little integration between Email and defect and requirement management tools (*like QC*). Teams had to rely on Excel or custom reporting tools to compile the relevant information manually from QC and then share it with the team. It was not uncommon (*and not even now*) to find long outlook threads discussing around a particular work item (*Bug, Requirement etc.*). Team members were obviously expected to check QC on regular basis to check the items assigned to them and take appropriate action. As expected, this wasn't real time and unless an email is sent to draw someone's attention there was latency.

### **PRESENT: Integration between Email and Tools like QC**

Not surprisingly, more and more add-ins were written when it became obvious that tools like QC need to talk to the people and that's where email notifications/subscription based reports became popular to send email notification to intended audience for faster action and response. This was definitely useful and solved the problem to a great extent and they are good to notifications/reporting scenarios but again not ideal to have discussions with having multiple team or entire teams copied (*and rightly so, because such conversations over email, if they are frequent are sometime discouraged as overhead and counter-productive instead*)

## FUTURE: Integration between Microblogging and Tools like QC

I think by now you would have guessed where we are heading to.



Practically teams don't live inside tools like QC which are foundation of our software development life cycle. They are humans and humans like to interact with humans on a platform that is designed for that very purpose.

That's the very reason we started making use of emails (*though unintentionally; initially manually and later on with the help of add-ins for email notification*) to ensure that the work items gets the due attention they deserve. Now with the changing times, emails are becoming obsolete (*not totally but when is the last time your wrote an email to your friend to catch up after creating your new Facebook account*) and people are more accessible on social networking /Microblogging site where you can have a quick discussion / chat with your friends (in our case team members) transparently in a more collaborative mode without spamming and swarming people with emails.

*We need our software processes & tools start talking to teams using a platform like Microblogging which is more apt for interacting with people because "Assigning a bug to a developer or raising a new CR might not serve its purpose unless it is communicated to the concerned team member and acted upon at the right time."*

Now let's quickly compare Email vs. Microblogging in context of project communication and try to find out when you should use what based on their respective advantages /disadvantages

## Comparison between Emails & Microblogging for Project communications

<b>Features</b>	<b>Email/Notification</b>	<b>Microblogging</b>	<b>Comments</b>
<b>Group discussions to come to a consensus or clarify items that need</b>			<i>Microblogging tools are better platform to have live discussions that involve team and quickly arriving at a decision. They can be sometime as good as call or</i>

<b>team involvement</b>			<i>live meeting with the ability to discuss with people connected at the moment and offline comments on the thread from others when they become online.</i>
<b>Detailed analysis and reporting</b>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<i>When there is need to share a detailed information to team like reports or metrics then email is much more elaborate and hence preferable</i>
<b>Instant report/ Point in time status</b>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<i>When you want to give a quick point-in-time update either Email or Microblogging can be used but the latter would be quicker and informal and should be used based on team's discretion</i>
<b>Information overload (Junk /Spam)</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<i>When teams start using emails to share project information quite frequently it can be very easily perceived as spam &amp; counter-productive whereas Microblogging can be proved more apt in those scenarios</i>
<b>Transparency</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<i>Emails are typically meant for sharing information with specific set of people (more personal in nature) and hence it's not the best choice to share information with the team that should be shared with everyone. Lots of time in fact unintentionally, people are not marked in important emails and don't get the information until it becomes too late.</i>

<b>Official &amp; Formal communication</b>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<i>Emails are best suited for officially communicating news/information to team that can be considered final (like approval/sign off)</i>
<b>Sharing confidential or sensitive information</b>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<i>Obviously if there is any project information that are sensitive then based on the sender's discretion, emails are still the way to go.</i>
<b>Voting / Showing consent or discontent with visibility to team</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<i>Though emails support voting but Microblogging are more intuitive and democratic way to show team's consent or discontent (thumbs up/thumbs down) by for instance liking or commenting on a subject which is cent percent transparent and visible to team</i>
<b>Instant communication (Real-time)</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<i>When you want to share something critical without much time spending drafting a neat email then Microblogging can be much more effective in serving the purpose.</i>
<b>Human factor</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<i>People love spending time on Microblogging as they feel more connected to friends and colleagues there as compared to using emails as communication platform.</i>
<b>Collaborating with customer</b>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<i>You can use both emails and Microblogging to connect with customers but</i>

<b>and stakeholders</b>			<i>lot of times customers / stakeholders might feel skeptical about sharing information on email as they may be later held accountable for it whereas on Microblogging they will be more open and willing to express their views/opinions and that is the very need of the agile to welcome and incorporate change.</i>
<b>Maintaining single repository of project discussion which can be later on searched and filtered</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<i>Emails are not best for searching and mining useful information about the project whereas Microblogging due to its nature with the help of features like tags can be quite useful to search or filter important informal discussion about a project for better understanding, analysis and post-mortem</i>

In a nutshell, we pay prices for using Email everywhere. Whether looking for information, clarification or seeking action, team members choose to send a message to Project DL instead of posting a question on a team's SharePoint Message board. This simple action instantly replicates their message in every recipient's mailbox and interrupts work of those who don't use Exchange rules and happened to keep Outlook open. Same happens with every answer including infamous and quite frequent "reply-all" avalanches. Eventually the thread winds down, satisfies the need of the person who asked the original question—and get erased from mailboxes or archived. A week, a month or half a year later the same question gets asked and the history repeats itself.

## Gap: Need integration between our Microblogging platform & QC like tools

In order to use Microblogging in conjunction with tools like QC, there is a need to come up with add-ins that can make the collaboration and experience much richer by seamlessly integrating the two. This section gives a glimpse of some scenarios where this marriage between the tools might prove fruitful.

### TFS to Microblogging Automatic Push

With the help of *TFStoMicrobloggingAddin*, which is configured that whenever:

- ✚ A new S1/P1 bug is opened by QA team then a *message is automatically posted on the OfficeTalk/SharePoint with the Work Item ID, title of the bug, severity and a tag to the name of the developer assigned*. This is of course visible to entire team and will act as a quick notification to the developer assigned
- ✚ A S1/P1 bug is Rejected or Declined by the developer, *a message is automatically posted on OfficeTalk/SharePoint with Work Item ID, Title, Reason and a tag to the tester assigned* so that he gets notified about the reason for failure and the discussion can start.
- ✚ A new Product backlog item or Change request is created for the current sprint or release, *then an automatic message is posted on OfficeTalk/SharePoint with Work item ID and Title* for the visibility of the entire team to discuss and comment on the scope

*Note: This was to give you the gist of the idea. The automatic configuration rules and the content of the post have to be flexible and custom based on a project need.*

### Microblogging to TFS Manual Push

Whenever a discussion happens about a work item on your Microblogging site then there should be a way to post back the discussion as part of Work Item history for future reference

## Prospect Scenarios leveraging Microblogging

*Fable: The team is working on a new version of an ERP from 4 different geographies. The customer, developers and testers are distributed across continents and work across overlapping time zones. The team is following scrum methodology. Team has adopted an internal Microblogging tool for project communication with TFS integration for faster collaboration.*


We are going to now look at few typical day-to-day activities of this team leveraging Microblogging to see how it's shaping up.

### Requirements:

#### New Product Backlog/Enhancement for a current sprint

- ✚ Wassim, a Business Analyst from Dubai on the team, adds a new Enhancement/Product Backlog item in QC as part of the current sprint
- ✚ TFStoMicroblogging Add-in is configured to *send an automatic message on Microblogging tool with the Work item Id and the Title*
- ✚ The conversation kick-off with this and the team and customers get involved in arriving at a decision by interacting with each other. At the end of the discussion, Business analyst can attach the entire conversation to back to work item history for future reference.

**ProjectTalk**



1022: As a user I should be able to scan the document





**New Product backlog item added few moments ago**

Automatically posted by QC Add-in!!

Wassim (Business Analyst)

*Developer X and 3 more people like it*

**Comments:**

Jack (Developer)		Why can't they use the soft attachments? Is it really high priority?
Wassim (Business Analyst)		Because this is important feature as per their users and it exist in their legacy system
Mark (Customer)		Team, this is really high priority and if required we can push out other low priority story to get this done
Jack (Developer)		Let's do it then.


Attach discussion to Work Item History

## Bug Lifecycle

### Triage Scenario: LIVE Bug triage for Critical and High Priority Issues

- ✚ Michelle, a test from Redmond discovers and logs a S1/P1 bug in TFS which needs immediate attention of the Gunjan, a developer from India
- ✚ TFStoMicroblogging Add-in is configured to post an *automatic update with Work Item Id, Title and a tag for Kate*
- ✚ This conversation kicks-off the conversation. At the end when the work item is marked as CLOSED, add-in is automatically configured to attach the entire conversation to back to work item history for future reference.

**ProjectTalk**



[#Kate1044](#): HRO app is crashing while launching [#buildfailure](#)


**New Severity 1 bug is created and assigned to you 2 mins ago**

Automatically posted by QC Add-in!!

Michelle (Tester)

*Comments:*


Gunjan (Developer)



I am in an all-day training. [#Jack](#) can you see this?


Sent from Windows Phone 7

Jack (Developer)



Looking into it as we speak


Jack (Developer)



[#Michelle](#): Work Item 1044 is marked as FIXED (Jack's comment: Library reference was missing). Change set Id: 2322

Automatically posted by OC Add-in!!

Michelle (Tester)



[#Jack](#): Work Item 1044 is marked as CLOSED (Bug Verified). No further action required

Automatically posted by QC Add-in!!

### Triage Scenario II: By-Design vs. Valid Bug debate (it's the way it is)

- ✚ Michelle, a tester reported a Severity 3 bug in TFS and assigns it to Gunjan
- ✚ Gunjan checks his TFS and notices the bug assigned to her and she feels it's By-Design and update the status along with her comments
- ✚ TFS to Microblogging add-in is configured to *post an automatic update on Microblogging tool with Work Item Id, Reason/Comment and tag for the Tester to whom it is assigned back.*

- ✚ This triggers another conversation and at the end of it. Using the push button the entire conversation is sent to Work item history for future reference

**ProjectTalk**



**#Michelle1054:** User can't select multiple records in one click (Status Updated: By-Design, Comment: As per specification)

**Bug is marked as BY-Design and assigned to you 5 mins ago**

Automatically posted by QC Add-in!!

Gunjan (Developer)

*Comments:*

Michelle (Tester)		Gunjan, it was discussed in our last review meeting with customer. <a href="#">#SAM</a> ,any comments
Wassim (Business Analyst)		Yes indeed. It's important to have multi-select checkbox. I have update Product backlog. Please check
Gunjan (Developer)		I see. Thanks.
Gunjan (Developer)		<p><b>#Michelle: Work Item 1054 is marked as FIXED. Change set ID: 4032</b></p> <p style="text-align: right; color: blue;">Automatically posted by QC Add-in!!</p>

Attach discussion to Work Item History

### Triage Scenario III: Bug Repro Issue (But it works on my machine)

- ✚ Michelle logs a Severity 2 bug in TFS and assigns it to Jack
- ✚ Jack notices the bug assigned to him in TFS but he can't repro the issue. He changes the status as Can't Repro and assigns it back to Michelle
- ✚ TFStoMicroblogging *Add-in* is configured to send an automatic update to Microblogging tool with Work Item Id, Status, Comment, and a Tag to name of the Tester

- ✦ This conversation kicks-off the conversation. At the end when the work item is marked as CLOSED, add-in is automatically configured to attach the entire conversation to back to work item history for future reference

**ProjectTalk**



**#Michelle**3354: Record is not getting saved (Status Updated: Rejected, Comment: Works on my machine every time☺)

**Bug is marked as Reject and assigned to you 5 mins ago**

Automatically posted by QC Add-in!!

Jack (Developer)

*Comments:*

Gunjan (Developer)




Jack, Michelle won't be online for another 4 hours. In the meantime I quickly checked on my machine and it doesn't work either unless you have DTC Windows service running

Sam (Business Analyst)




Ah. So it's a configuration issue

Michelle (Tester)



Cool. Let me retry

Michelle (Tester)



**#Jack**Work Item 3354 is marked as CLOSED. No further action required

Automatically posted by QC Add-in!!

## Instant Reporting & Monitoring (fostering transparency)


### Management self-reporting

- ✦ Manager notices from the reports that the team is missing important milestones due to frequent build failures
- ✦ He goes and filters all the post that are tagged for build failures


- ✚ By going through these post / discussions related to build failures he understands the root cause and finds a pattern

*Note: Apart from status reports (metrics), the Manager is now empowered and equipped with the ability to see the ground reality himself and drill-down deeper to help the team more effectively by culminating the root cause.*

**ProjectTalk**




[#buildfailure, #Severity1Issues](#)



Greg (Manager)


*3 conversations found*

Michelle (Tester)




**#Kate1044:** HRO app is crashing while launching


New Severity 1 bug is created [#buildfailure](#)

 [View 3 comments](#)


Sam (Business Analyst)



**#Jack323:** Update is deleting the data. [#buildfailure](#)


 [View 5 comments](#)

Michelle (Tester)



**#Kate1044:** Configuration file reference is missing&app is hanging

New Severity 1 bug is created [#buildfailure](#)

 [View 3 comments](#)

**Customer involvement & participation (truly)**

- ✚ Customer receives his usual daily burn down charts and he notices there is sharp dip in the team's velocity over last 3 days
- ✚ He goes to Microblogging tool and filters the project conversation based on the relevant tags to better understand the current affairs that might be the culprit
- ✚ It is evident that there were multiple clarifications on the requirements/user stories and team is unable to make expected progress as a consequence
- ✚ He picks up his phone to call the product owner and business analyst to understand the reason and help the team

## Pros

### Productivity improvement

Microblogging's integration with QC like tools will result in faster resolution of issues, help the team to quickly arrive at a consensus and ensure that important information is communicated more effectively etc., which will result in more throughput by the team.

Apart from being effective this will also make the team more efficient by triaging for instance on Microblogging can eventually cut down the effort and the time spent on meetings, calls and emails as the conversations on Microblogging can take place without the need to have everyone online at the same time.

Your team members can respond when they are away on lunch from their shiny Window 8 Slate or from Window Phone 7 when at home without the need to connect to TFSWebAccess

### Making your communication more agile

Microblogging is instant whereas emails are relative slower in responsiveness so it can be faster to have real-time conversation over web without the need to make a phone call or requiring physical presence.

### Fostering Transparency

As an honest company, we want to encourage transparent flow of project information to everyone in the team including our customers,

partners and stakeholders to reduce any communication related issues. Microblogging is much more effective than emails by eliminating the chances of important communication getting lost or tampered that often leads to different versions of truth.

## **Customer and Stakeholder involvement on live basis**

### **(Virtual presence)**

It can be mutually beneficially to have your customers be part of your project conversations. One, it can help customer and stakeholder get a feel of what's happening in the project real time and two, they will feel more involved and really the part of the team. On the other hand, they can provide your constant & valuable early feedback in accordance with agile philosophy.

## **Live monitoring and Subjective Reporting**

### **(Yes it's not all about numbers always)**

Stakeholders and Management's involvement or silent role in project conversation will not only foster healthy and transparent culture but will also act as instant reporting without metrics. This can't and won't replace daily/weekly reports with metrics but lot of times this can provide them with a cognizance by following the project status update and conversations that can't be alone obtained from plain metrics.

## **Reducing 'Overload of Information' syndrome**

Microblogging is good to keep the discussions short and crisp (*remember 140 words limit to convey your messages*) and consciously putting an end to overload of information. Lot of time we can communicate the message in fewer words but fail to do so because of the habits.

## **Way to go (Its more human)**

The fact is emails aren't cool anymore. People are addicted to Microblogging and they literally sleep, eat and drink there and this will connect more with teams than traditional emails.

## **Things to consider**

### **Well, it's not a replacement (so it's not that big a change)**

It is not going to replace your existing communication tools be it email or notification or subscribed reports but instead it's going to find its own place and will be used in situations and scenarios which are ideal candidate for Microblogging.

### **Decide the granularity**

Well, when you use such a Microblogging team for Project communication, you as a team need to decide this and configure the tool accordingly . *For instance you can configure a rule that post an automatic message on behalf of the creator whenever a Severity 1 bug is raised (Posting every bug will dilute the purpose and will make it less impactful)*

### **Management buy-in & Team commitment**

Just like any other major change in the way we do business, it can't be done easily when you have management's buy-in and support. It's important to take them aboard.

It's not a bad idea to start with a POC and let the team feel the advantages of it and fix the loopholes along the way and then you will have more people on your side and smoother adoptions.

### **Needs Discipline**

Once you have management and team enrolled in, you need to come up with a set of protocols (they are not strict rules but guidelines) which your team should try to adhere to. It's very easy to lose track and later on blaming the idea was after all not-such-a-good-one.

### **Need investment on customizing tool (Add in)**

The reality is that you need to have Office Talk/SharePoint deployed, customized and integrated with your QC like tools. This will need community contribution and investment in coming up with those Add-ins.

## Closing Thought

If the readers and our esteemed reviewers feel that this can be progressive changes we should embrace then we need your support and we are calling for actions to take this idea places.

## Author's Biography

Raj Kamal is a Senior Test consultant with 9 yrs. of experience, specializing in different types of testing techniques, test automation and testability in different domains like Manufacturing, Healthcare and Higher Education. He holds an APICS certification in Supply Chain Management. Familiar with Rational, Mercury & Microsoft testing tools, he has helped teams develop test automation strategies and architectures for such companies as Cognizant Technology Solutions, Oracle Corporation & Microsoft. He also provides training in automated testing architectures and design. He is QAI(CSTE) & ISTQB Certified. He has a master's degree in Computer Applications. He is currently working as a Test Lead at Microsoft, India, Business Intelligence domain. His passion: <http://geektester.blogspot.com/>

Past presentations:

- Presented a webcast at Microsoft worldwide customers for [Data quality Testing](#)
- Represented Microsoft at QAI Software Testing International conference, Florida as a speaker "*Application and Script Independent Automation Framework*" (<http://www.qaiworldwide.org/qai.html>)
- Represent Oracle Corporation as Speaker on using "*Mapping Rational Unified Process with Rational Testing Tools*" at International Forum for Software Testing Professionals <http://ifstep.org/>
- Represented Microsoft as a speaker at [Test 2008 International Conference](#)
- Automation framework designed is accepted for Presentation and Publishing at [International Conference on Information Technology: New Generations](#), Las Vegas & IEEE Digital Library

Publications:

- Published a paper on [Resurrecting the Prodigal Son--Data Quality \(Stickyminds\)](#)
- Published a paper on ["Adventure of BI/DW Testing" \(Stickyminds\)](#)