Brian Braddock

“I want this team and project to be successful and I will type my figures numb to make sure it happens, but is hard to know where I can be the most helpful”

**Age**: 32 **Location**: DTH, MI **Tech Exp**: Very High **Title**: Software Engineer

**About Brian:**

Brian is considered a “Subject Authority” on the team due to his time and experience with one or more of the components that make up MyApps. As part of this unofficial responsibility Brian happily spends the majority of his time researching the feasibility of business needs/requests or possible improvements to the project or the way the team works. The rest of the time he is mostly working independently on technical tasks which may not have a clear path between the current state of the system and the desired state. Brian keeps his own mental backlog of “5 minute” tasks to do as he gets time to complete them. When he has longer to spend on a task he tries to help others who might be road blocked or having a hard time with their task.

Generally speaking, Brian doesn’t have a problem staying busy, but he is concerned that he is allocating his time and talent to the most important tasks that need to be completed in order to meet MyApps business commitments and the goals for the product’s future. He also don’t have much difficultly getting started on new tasks as it is not uncommon that he was the one who identified that new work to begin with.

Brian is a team player who will always strive to work in a way that benefits the team and the project, even when it might come at the cost of his own personal interests. He is very approachable and is always willing to have a conversation, pair programming session or just a causal conversation when he has the time to do so. Brian feels a responsibility to help others find their way through the complicated technical landscape that is GE and will often be found mentoring those he can.

A graduate in Computer Science, with 10+ years of technical experience Brian is widely recognized at the site and within CoreTech for his technical expertise. A proud father and beer connoisseur, Brian always has a good story to share when there is a lull in the action.

**Behavioral Considerations:**

* A big picture thinker who contributes to high-level project direction and implementation
* Self-starter who needs very minimal information to get started and feel comfortable
* Experienced with the project architecture
* Feels pressure to correctly allocate time between responsibilities and show progress against project work
* Doesn’t feel like non-project work is considered progress
* Comfortable independently talking to others and researching to determine a set of steps, which he is confident align with the project high-level goals and fulfill the requirements of the task.
* Conducts most planning and status updates through verbal, 1x1 conversations.
* Feels comfortable and experienced enough to make suggestions or changes around development processes and tooling

**Goals:**

* Have the information he needs in order to empower him to make the priority decisions he is trusted to make
* Wants to know what major initiatives/releases the team is working toward, what is done, what we need to completed and when it is all due
* Wants to complete his tasks in a way which supports the high-level goal they are related too and will be flexible enough for future requirements
* Wants to know he has prioritized work in the right way in order to be confident he is doing the right work
* Wants to know that the non-project work he is doing matters too
* Wants to be able to jump in and help out on other’s tasks when required/available to do so or when they are blocked and he can help.

**Pain-Points:**

* Not getting the information needed to make priority decisions between project work, research, firefighting, automation, internal process improvement, mentoring and collaboration.
* Standups/Planners are solely focused on low-level and tactical conversations and updates which do not help give a clear picture of the project needs at a high level
* Finding what he can work on next
* Seeing who is blocked and why
* Time that is spent researching a task that is not tracked
* Lack of visibility into the work that he or others are responsible for not related to MyApps
* Not seeing the relationship between the tasks that he is working on and what others are doing, which can cause one to be a blocker for the other.

**Needs:**

* See Non-Project work such as Fire Fighting efforts, Research, Internal Process Improvements and others tracked and shown as progress
* Determine the priority of each task in the ready state he could work on and see what initiative it maps up to/is a child of
* Separate High Level Major Initiative conversations to cover each initiatives status and how current tasks relate to the success of the initiatives.
* Quickly find the relationship between task and who is the authority behind it
* Task tickets with a basic amount of information to point him in the right direction to start research & implementation
* A quick easy way to update the status of his tickets which feels as easy as stating it in a standup
* A medium to discuss the outcomes of personal research around internal process improvements or interesting research with interested/relevant parties
* Easily visible plan that includes release dates and expected features/functionality/fixes
* Easily see progress toward a release/initiative
* Prioritize and relate what we are developing to the release goal

 Miles Morales

“This place is like no where I have ever worked before and the challenges we overcome everyday could only be found at GE”

**Age**: 29 **Location**: DTH, MI **Tech Exp**: High **Title**: Software Engineer

**About Miles:**

Miles is a talented developer who joined the MyApps team last year after spending three years with an automotive company developing as part of an agile team creating a high visibility IT infrastructure project. He is used to the struggles of being a young forward-thinking professional working in a traditional IT environment and is happy that he was able to find a pocket of like-minded individuals inside of GE to escape the normal IT mindset.

Compared to his previous job, Miles is in awe of the environment he now enjoys working in as part of the MyApps Team. Never before has been trusted with this level of independence to prioritize and implement his work as he deems necessary to continue MyApps success. Despite his joy around the way the team and his leadership trusts him to do the right work the right way, he is worried that he doesn’t always have enough insight into what more senior members of the team have planned for the product and what business needs leadership is anticipating the product will have to address in the future. He gets most of this information during lunch, as it is when some of the most valuable high-level conversations happen. For Miles it is an unscripted opportunity to hear about what everyone else is working on.

While Miles is no doubt a smart guy, MyApps is by no means a simple system and Miles still feels like he is just getting to understand the different ways each part of the solution works together to provide secure remote access to the user. He is thankful to be able apart of a very approachable team of smart individuals who are willing to help him create reasonable implementation plans for some of the harder problems he is tasked with solving.

A graduate in Computer science, Miles is able to hold his own in any general technical conversation, but finds he sometimes struggles to translate that technical know-how to GE technical know how; given his limited time with company. As a result Miles is sometimes seen as the talented, dependable, but quiet engineer who is making a name for himself through it consistent results.

**Behavioral Considerations:**

* Wants to have a plan for how to implement the solution for a task when leaving standup or planning so he is confident he is doing it in a way that is consistent with the team’s approach and supports high level project goals
* Does not feel like he has the experience to contribute in “this is what we should do and how” types of discussions (planners)
* Focused more on implementation rather than research and discovery
* Rarely comes up with new features or process improvements for the team to discuss
* Is willing to do some independent research to determine an implementation plan, but would rather speak to an authority on the team to design a solution.

**Goals:**

* Quickly see what the team is working on and get a clear picture of how all that work interacts and will effect the system
* Implement his solutions in a way, which supports the future plans for the project and the team’s expectations.
* Keep up with what is coming down the pipeline for the project and changes to the way he will work on it
* Continue to learn more about the product so he can become more involved in future planning sessions

**Pain-Points:**

* Doesn’t have a lot of visibility into what others are doing
* Doesn’t always know how the work that others are doing fits into the project and relates to the work he is doing
* Not having a clear implementation plan he knows will meet the needs of the team and the product before leaving the planner
* Not having a clear idea of what the future of the team, project and org is
* Having to rework a solution because he had to develop something that was impacted by a future-thinking discussion he wasn’t in.
* Being stuck for days waiting to talk to a team member about an issue he needs help with

**Needs:**

* Map backlog work to the skills he has in order to see where he would be the most useful
* Make priority decisions based on pre-established priorities the team has already made for a set of features
* Opportunities to get Subject Authority weigh-in on a solution
* High-level unscripted conversations about the future of the team, org and product
* The ability to have1x1 conversations with teammates in a reasonable amount of time
* To know the availability status of teammates in terms of can talk, can help, can’t talk, is fighting a fire, etc.
* To know what others are brewing on
* Understand how the different components of the project work together

 Reed Richards

“It’s like we are cheating, but really we are just following a set of principles which has already been shown to be successful”

**Age**: 37 **Location**: DTH, MI **Tech Exp**: High **Title**: MAA Leadership Team

**About Reed:**

Reed has earned a reputation as a positive, but disruptive force in GE and the technology he represents shows. It is a reputation, which is not by accident and has been carefully crafted over his seven years with GE. Additionally it is a reputation, which he protects by consistently reevaluating his team’s direction and its way of doing things in order to make sure they are always in the right conversations. In these conversations Reed does his best to represent his team as accurately as possible, which requires that he be involved enough in their day to day in order to do just that. In order to accomplish that level of involvement he makes sure to be careful not to step over the line and become so involved that his team feels they are not trusted to innovate and do their jobs.

For Reed this is more than just a job, it is a responsibility to those who he works with to make sure they have every opportunity to succeed within the Big GE he is constantly holding at bay. This personal stake that he takes in his work and the relationships he builds with his team is the fabric that allows the team to be successful in a broader environment where a team like his might not otherwise be able to exist. The flip side of that personal stake and close relationships is the pressure he feels to succeed. As such Reed constantly feels the need to make sure he prepared for the next critical conversation that could impact the team.

Reed is a Computer Information Systems graduate who has the heart of a tinkerer. When the workday is done, his next one begins, building his own contraption or helping his kids with theirs.

**Behavioral Considerations:**

* Dedicated to the success of the product his team represents and the success of his team
* Has worked hard to earn his reputation and will continue to work hard to keep it
* Tries to be as forward thinking as possible
* Does the best he can to have a good balance between trust/hands off and involvement in project development
* Has a personal stake in the outcomes of the project
* A visionary who tries to be two steps ahead, creating a roadmap to meet anticipated future business needs

**Goals:**

* To be able to see roadblocks engineers might not explicitly state
* Be able to accurately predict how long a feature will take to develop and make those predictions based on what the team has achieved in the past (cycle time)
* Have a clear picture of what is being developed today, its progress and when it will roughly be completed
* Have a clear understanding of the impact of what’s being developed and the impact of any risks/issues/bugs.
* State this is when we can get to some new business requirement and here is how long to will take to complete with confidence (lead time)

**Pain-Points:**

* Not being able to have informed priority, feasibility timeline discussion with MyApps stakeholders and business partners due to unreliable, inaccurate, out-of-date team performance and progress information.
* Not being able to see the business impact of the work the team is doing
* Having to have 1x1 conversations to find out how close we are to completing something
* Tasks are too large to be useful to track and involve more than one person

**Needs:**

* Know within orders of magnitude how much longer in-progress tasks will take without talking directly to an engineer
* Derive business impact for a task based on the ticket/card or the release plan
* Be able to see a task move through our progress in a reasonable amount of time, not over the course of months
* See what open issues the team is addressing and the scope of the effect they are having on the system

Takeaways, Observations and Learning’s

“I violently agree!”

**A Story of Trust and Empowerment:**

This team has an environment of trusting everyone to take the initiative to do the right work, in the right order, in the right way to support the team and the success of the product, but we are struggling to have the information we need in order to make the technical implementation, priority and forward thinking decisions we have been empowered to make.

* We trust each other to do the best work possible in a reasonable amount of time, but we don’t share the forward-thinking knowledge and business impact understanding equally across the team that is required to make these types of decisions.
* 30% of this information doesn’t exist
* Sometimes we trust too much – don’t like the decisions – sense of initiative and care about the product can sometimes translate to “I am going to do what I think is best”

All facets of the team are starved for the information they need to make the decisions they want to make today in order to do their work in the best way possible

* Dev team isn’t telling QA what they are doing and their progress in enough detail that QA can create accurate/reliable test plans
* Leadership/Subject Authorities within the team aren’t spreading an understanding of the proposed future of the product to the Dev team in order for them to develop their solutions in such a way which will support the future state and correctly prioritize work with confidence
* The Dev Team doesn’t communicate the business impact of the work they are doing and it’s estimated complete date to enable Leadership/Subject Authorities to accurately represent the team
* The team doesn’t provide a clear, trustworthy picture (in orders of magnitude) of when it is reasonable to complete a business need, based on previous work, so that Team Leadership can prioritize it with other [business] needs that exist.
* The Dev Team within itself doesn’t know the relationships between the tasks they are working on and roadblocks other engineers might be experiencing.
* The entire team doesn’t know what others are brewing on that could effect the future of the project and the way that we work.

**Recommendations:**

**Team Member Office Hours**

Set aside an agreed upon amount of time (ex an hour a day) where team members are no questions asked available for conversations about anything.

**Addresses:**

Long waits to talk to team members

Not knowing if team members are available to talk

Not aware of what other team members are doing

Identifying roadblocks

Opportunities to talk about future of the product/project

Guilt-free collaboration opportunities

Non-scripted conversation opportunities

**Team Member Status Indicators**

A detailed virtual indicator showing how available a team member is for a question, conversation or collaboration session.

**Addresses:**

Long waits to talk to team members

Not knowing if team members are available to talk

Not aware of what other team members are doing

**“Fireside Chats” with Stephen**

A reoccurring scheduled time set aside for Stephen to talk with the team about the direction of the team, product, org or home brewing.

**Addresses:**

Inconsistent distribution of future-thinking/organization information across the team

Not aware of what other team members are doing

Challenges prioritizing work

**“Virtual Lunch/Breakfast”**

A reoccurring scheduled meal where the entire team eats “together” to encourage non-scripted conversation about anything between team members.

**Addresses:**

Inconsistent distribution of future-thinking/organization information across the team

Not aware of what other team members are doing

Opportunities to talk about future of the product/project

Guilt-free collaboration opportunities

Non-scripted conversation opportunities

**Track Reasonable Non-Project Work**

When there is meaningful internal process improvement research, proof of concept research, automation work, firefighting, etc being done, track that work.

**Addresses:**

Not aware of what other team members are doing

Challenges prioritizing work

**Start Ambiguous Tasks with a Research Card**

When a new task does not have a clear implementation plan from the start create a research task, which will have a Definition of Done of outlining the tasks required to complete the new task.

**Addresses:**

Tasks are too large to be meaningful

Task progress is hard to determine

Dev team doesn’t feel like research time “counts”

**Translate Research Cards into Child Tasks with Research Label**

Use the output of a research task to create reasonable sized and meaningful child tasks, which are relatable back to the original research task.

**Addresses:**

Tasks are too large to be meaningful

Task progress is hard to determine

Team doesn’t know the relationship/interdependencies between what they are doing and what others are doing

Dev team doesn’t feel like research time “counts”

Challenges prioritizing work

**Detail Business Impact in Ticket**

Make it a requirement to detail the business impact doing the work included in a ticket will have

**Addresses:**

Representative Authorities for the team don’t know the impact of the work the team is doing

Representative Authorities aren’t sure what the progress is of important features/functions

Challenges prioritizing work

**Organize Tickets in a Release Plan View**

Show a roadmap view of upcoming releases with a prioritized view of major efforts the release might include. As the release gets closer the prioritized view of major efforts should become more fleshed out and show progress.

**Addresses:**

Representative Authorities for the team don’t know the impact of the work the team is doing

Representative Authorities & Team aren’t sure what the progress is of important features/functions

Challenges prioritizing work

Challenges relating tasks to high-level efforts

**Have a Firefighting/Bug Work View**

Show what the team is working on fixing, the business impact of the issue and the progress made so far.

**Addresses:**

Representative Authorities for the team don’t know the impact of the work the team is doing

Representative Authorities aren’t sure what the progress is of important features/functions

Team doesn’t know what others are doing

Challenges prioritizing work

Team doesn’t feel like firefighting/bug work “counts”

**Make Roadblocks Clear**

Show when a team member is road blocked and clearly state what they need in order to overcome the block

**Addresses:**

Representative Authorities for the team don’t know what team needs to overcome road blocks

Team doesn’t know what others are doing

Team doesn’t know who is road blocked

Proposed Tool Features

“Estimate, Impact & Communicate All The Things”

Team Member Status Indicators

* Firefighting, Heads-Down, Can Chat, Can Collaborate

Bug/Firefighting Area

* Impact of issue, progress resolving, steps taken

Non-Project Work Card Types

Research Card

* Becomes parent card for implementation or further research tasks

Progress Indicator on Tasks

Business Impacts for Tasks, Epics, Releases and Bugs

Push an Alert for a Request for Help

Road Blocked

* Include what is needed to be unblocked

Show a Release Plan View

* Show tickets associated with each feature in release
* Prioritize the tickets
* Show anticipated features in future releases