

Cost Allocation - Retro: 8/6/2020

@diego, @gonzalo, @marilyn, @andres, @braulio

Ideas generales

- **@andres**

- Martin llevo el producto a un lugar que no fue el correcto. Tuvo una visión parcial. Martin tomó el rumbo equivocado.
- Rohit puesto por Vincent para sanear el error de Martin, de alguna manera salvar el proyecto.
- Rohit es developer (deformación profesional). Busca detalle de código.
- La idea de la participación de @andres es ser fusible para con Rohit.
- Se necesita armar el caso con la información de todo el equipo de OA.
- No hay un rol que cumpla la función de PM en el proyecto Cost Allocation.

Que no funcionó

- **@marilyn**

- Se respira mucha presión, se piden muchas cosas que no son necesarias para el momento del proyecto que se esta. Esto se acrecentó desde el momento en que Rohit comenzó a participar en el proyecto.
- Rohit hace Dailys eternas. Utiliza una Daily para sacar todo tema que le preocupa el proyecto perdiendo el foco de la reunión.
- El proyecto cambia el scope constantemente. El negocio no esta seguro de lo que se necesita. Y muchos métodos de calculo se cambian de cuajo.
- El MVP no es tan un MVP, sino se esta reemplazando un producto completo. Se esta llevando un enfoque de cambio de producto completo a algo que debería ser un MVP o viceversa.
- Falta de tester, entonces los analistas testean y todo se hace en simultaneo. Se mezclan tareas y etapas del proceso de desarrollo.
- Se siente que Rohit intenta exponer a Martin en tareas que no puede cumplir y al mismo tiempo esas tareas recaen en el equipo de OA. Eso va en detrimento de la calidad de las tareas que el equipo puede entregar. Ya que se suman items que no estaban en el scope inicial de los Sprints.
- No se esta usando el Board de Jira de manera correcta. No se sigue un proceso de tareas/tickets. Muchas de las tareas se piden en slack o de palabra y no entran en el scope del Sprint. Se sobrecarga el equipo de OA.
- Se van agregando cosas en el Sprint, no se mantiene cerrado lo que se planifico en el Sprint Planning, siempre se siguen sumando cosas en el medio del Sprint.

- **@diego**

- El problema no es Martin, sino el problema es Rohit y constante agregado de tareas.
- Sostiene que Andres esta errando el diagnostico.
 - Ej: Rohit pidió determinar la cantidad de lecturas a Martin. Martin pide ayuda a OA porque no lo puede hacer y hace que Diego y Gonzalo tengan que dar soporte sobre eso, al mismo tiempo pidiendo que ejecuten los algoritmos de manera correcta. Es imposible hacer las 2 cosas al mismo tiempo. Muchas cosas en el mismo plato y todo es prioritario.

- Se siente desgaste al estar dando constantes explicaciones.
- Rohit no prioriza y cada tarea que suma entra como prioritaria sin determinar el impacto en las anteriores tareas.
- Nadie de XPO frena a Rohit, porque es el VP (palabras de Martin).
- Se suman cosas al release y no cambian las fechas de entrega del release.
- Rohit genera discusiones sin sentido sobre performance especificas, que si bien en algún punto aportan, se lo hace en reuniones que tienen otro scope.
- Se le pidió a Rohit que cada mejora se trackee en confluence y se priorice, pero sin éxito se pide todo junto en el momento.
- Rohit utiliza algunas meeting para conocimiento propio y no para el objetivo de la meeting.
 - Ej: se realizo una pre-planning de Sprint y en lugar seguir esa agenda de reunión se hablo gran parte de la reunión del release 3.
- **@gonza**
 - Rohit sabe que las cosas que le pide a Martin no las puede hacer y cae sobre los desarrolladores y eso hace retrasar las cosas acordadas para el Sprint.
 - Los tiempos de deployment son largos y se piden constantemente.
- **@braulio**
 - (TBD)

Action items

- Encontrar una persona que exprese a Rohit que el proceso no esta funcionando.
- Encontrar un moderador para las reuniones, por ejemplo Terri (PM) y que se respete el scope de la reunión y el timebox. El moderador debe ser alguien externo que pueda frenar las necesidades del VP (sin miedo a reprimendas) y en caso que haya necesidades fuera del scope de la reunión, debe armar una nueva reunión.
- Todas las tareas que se piden por mínima que sea debe trackearse en Jira. Sea: nueva user story, modificación de user story, nuevo modulo, modificación de modulo o **mejora de performance**.
- No deben agregarse tareas fuera de las planning al Sprint.
- Determinar Product Owner que priorice las tareas. Las tareas deben quedar priorizadas en Jira.
- Si se agrega algo al release, se debe extender la fecha o quitar otra tarea.
- **Falta de tester y el impacto en analistas y calidad del producto.**
- (TBD)