

Scrum @ TCS UY

Autor: Diego Garagorry

Fecha: 24/06/2010

Agenda

- Presentación del Proyecto
 - Desafíos
 - Organización General
 - Sprints
 - Resultados
 - Lecciones Aprendidas
 - Buenas Prácticas
 - Conclusiones

Presentación del proyecto

Proyecto

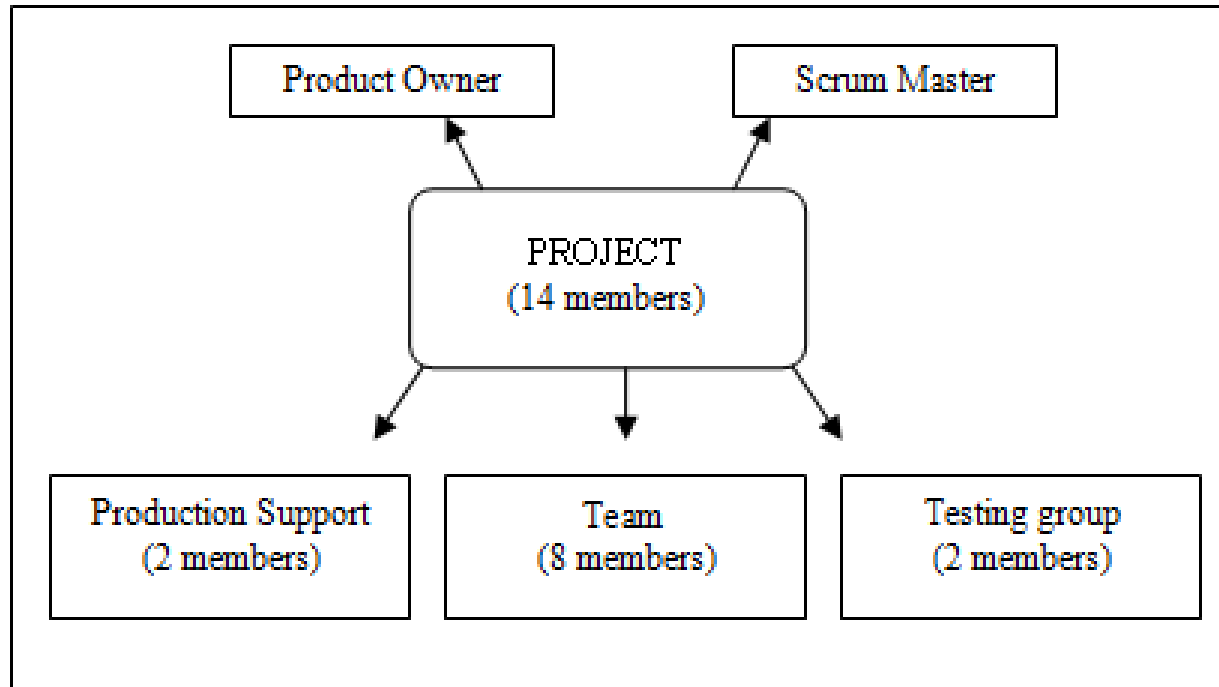
- Cliente: Empresa de seguros con presencia en más de 30 países:
- Objetivo: Crear una aplicación web simple e intuitiva que permitiera cotizar, emitir pólizas y crear certificados para seguros marítimos.
- Metodología propuesta:
 - Scrum con sprints de 4 semanas.
 - Realización de casos de uso durante sprints previos.
 - Construcción en módulos para sus distintas sucursales en iteraciones sucesivas.
 - Sprint 1 y 2: Módulo de cotización para el primer país
 - Sprint 3: Módulo de cotización para el segundo país.
 - Sprint 4: Módulo de administración y certificados para los primeros países.
 - Incorporación de cambios solicitados en siguientes sprints.



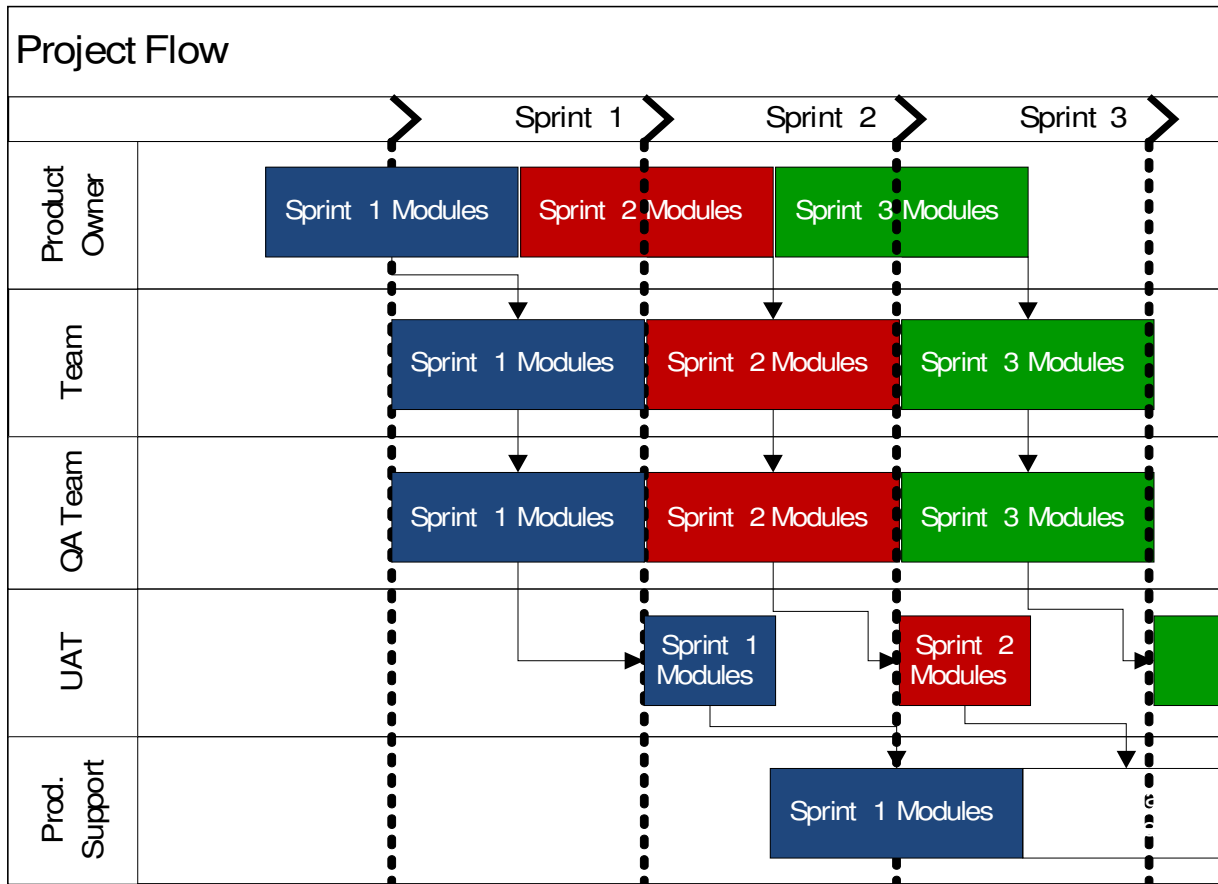
Desafíos

- Necesidad de salir al mercado en 8 semanas.
- Requerimientos definidos solo para las implementaciones iniciales.
- Calendario de implementación en los distintos países no definido.

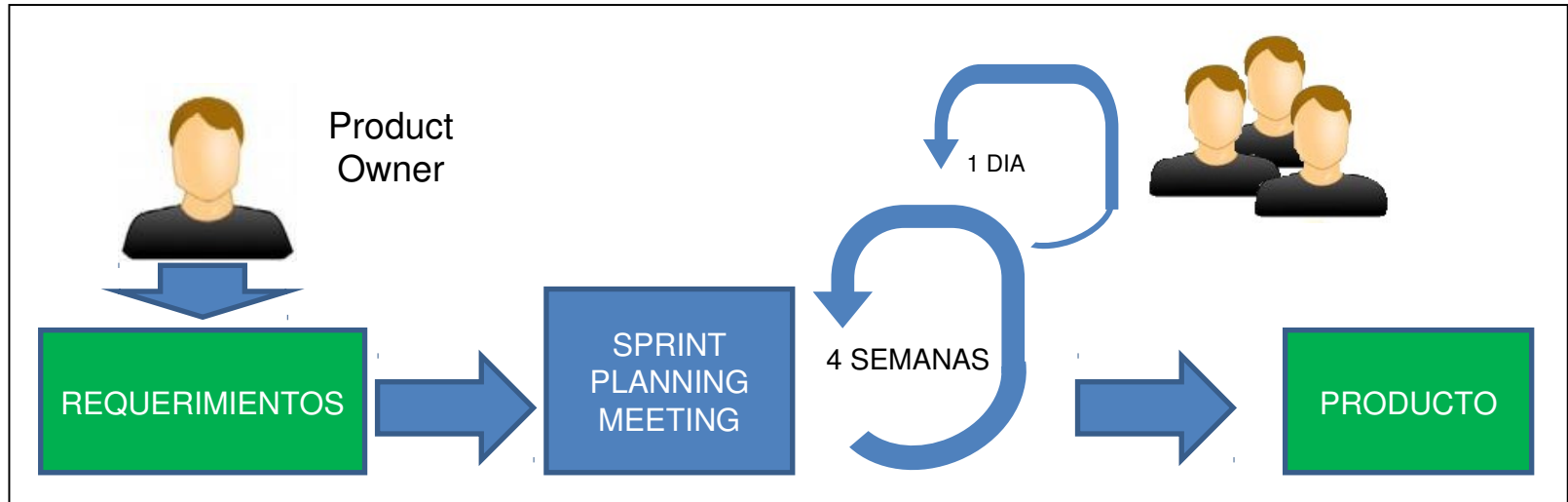
Organización General



Organización General



Sprints



Resultados

- Se cumplió con las fechas esperadas.
- Se evitó re-trabajo diseñando solo lo necesario.
- Se realizaron ajustes sprint a sprint.
- Buen entendimiento de requerimientos.
- Buenas estimaciones.
- Aumento de la productividad sprint a sprint.

Lecciones Aprendidas y Desafíos

- Es recomendable tener un Sprint 0 para preparar tanto la arquitectura inicial como los ambientes de desarrollo, definir tecnologías, etc.
- Al no tener una visión completa del producto final a desarrollar es necesario hacer mucho refactoring sobre la marcha. Es importante desarrollar pensando en eso.
- Durante cada Sprint el equipo no solo desarrolló si no que corrigió bugs de UAT teniendo que manejar varios ambientes en simultaneo. Esto debe considerarse en las estimaciones.
- Luego del primer Sprint las puestas en producción son sobre aplicaciones en uso por lo que generar scripts para insertar datos maestros en la base ahorra mucho tiempo.

Buenas Prácticas

- Complementar Scrum con prácticas de XP como integración continua, refactoring, tests de unidad y test funcionales automáticos. (TestNG, Qengine).
- Utilizar herramientas como Webex para mejorar la comunicación y realizar las demostraciones.
- Organizar reuniones entre el equipo de desarrollo y expertos en el área de negocios para asegurar el correcto entendimiento de los requerimientos.
- Proceso definido y claro para gestionar cambios.
- Uso de herramientas web como Bugzilla para bug tracking

Conclusiones

- Se cumplieron los objetivos del proyecto siguiendo metodologías ágiles verificando los resultados esperados.
- Se destaca:
 - Mayor velocidad de salida al mercado
 - Mejora en las estimaciones
 - Respuesta a cambios en mercado o requerimientos

Gracias

