



Universidad Tecnológica De Panamá
Facultad de Ingeniería de Sistemas Computacionales



Lic. Desarrollo y Gestión de Software

Gerencia de Proyectos Informáticos

Actividad Grupal Interactiva: Foro de Debate y Aportes

Grupo N.º 2

Profesor: Ronald Ponce

Integrantes:

Arroyo, Diego - 8-989-723

Sánchez, Isalleth - 8-1015-1027

Delgado, Einar – 7-714-1457

Barrios, Justin – 8-983-1021

Cáceres, Jorge – 4-827-858

Quintero, Estiven – 7-713-1048

Alonso, Ahmed – 6-727-783

Segundo Semestre

Año 2026

Contexto, preguntas y respuestas

1. El dilema de la dependencia en las Historias de Usuario bajo el estándar INVEST

Contexto: el acrónimo INVEST define las características de una buena historia de usuario. El principio de Independencia dicta que las historias deben poder completarse en cualquier orden, mientras que el principio de Estimación exige que los programadores puedan predecir de forma razonable el esfuerzo requerido.

Pregunta reflexiva: en sistemas de software complejos, la interdependencia técnica es una realidad común. ¿De qué manera la violación del principio de "Independencia" afecta directamente la capacidad de "Estimación" de las historias de usuario y qué impacto tiene esto en la conformación del Sprint Backlog y el cumplimiento del Sprint Goal?

Respuesta: En sistemas reales, muy pocas historias son totalmente independientes, porque suelen compartir una misma base de datos, un módulo de login o una misma interfaz. Cuando esa independencia se rompe, la estimación deja de ser confiable, porque el equipo ya no calcula el esfuerzo de una tarea aislada, sino el de una cadena de tareas que no controla del todo. Por ejemplo, estimar la historia "pagar con tarjeta" sin haber resuelto antes "agregar productos al carrito" puede llevar a subestimar el trabajo. Esto afecta al Sprint Backlog, porque historias dependientes pueden quedar bloqueadas a mitad del Sprint, generando retrabajo y poniendo en riesgo el Sprint Goal. Por eso, más que buscar una independencia perfecta, el equipo debe identificar estas dependencias durante el refinamiento del Product Backlog y planificarlas con anticipación.

2. El principio "Pull" de Lean frente a la planificación por lotes en Scrum

Contexto: la metodología Lean se basa en establecer un sistema "pull" donde se incorpora nuevo trabajo únicamente a medida que se finalizan tareas previas, para evitar la sobreproducción y mantener la eficiencia. En Scrum, el equipo selecciona cooperativamente un lote de temas durante el Sprint Planning y se compromete a completarlos en una iteración fija.

Pregunta reflexiva: ¿cómo se contrapone el sistema "pull" puro de Lean, donde el flujo es continuo y guiado por la demanda inmediata, con el enfoque de Scrum de comprometer un conjunto de tareas para un bloque de tiempo fijo (Sprint)? ¿Cómo puede Kanban actuar como puente para flexibilizar la rigidez de los lotes temporales de Scrum?

Respuesta: Lean plantea un sistema "pull": un nuevo trabajo solo entra cuando el anterior ya se terminó, en un flujo continuo guiado por la demanda. Scrum, en cambio, trabaja por lotes: en el Sprint Planning el equipo se compromete con un conjunto fijo de historias para un Sprint de duración fija, y una vez que empieza, "no hay cambios en un sprint". Esto le da previsibilidad a Scrum, pero le quita algo de la flexibilidad continua que sí tiene Lean. Kanban funciona como puente entre ambos: con su tablero visual y sus límites de trabajo en progreso o WIP, permite que dentro del propio Sprint las tareas se vayan "jalando" a medida que hay capacidad disponible, en lugar de arrancarlas todas al mismo tiempo. Así, el Sprint sigue siendo un bloque fijo, pero el trabajo dentro de él fluye de una manera más parecida a un sistema pull.

3. El límite de la facilitación del Scrum Master ante fallas estructurales

Contexto: el Scrum Master es responsable de asegurar que el equipo sea productivo, remover impedimentos y actuar como escudo ante interferencias externas. Sin embargo, factores críticos de éxito como el "soporte administrativo de alto nivel" y la "provisión de recursos" dependen enteramente de la alta dirección de la organización.

Pregunta reflexiva: si un proyecto se enfrenta a causas de fracaso estructurales como "recursos limitados" o "falta de apoyo de la dirección", ¿cuál es el límite del rol facilitador del Scrum Master? ¿Cómo puede este rol influir a nivel organizacional para resolver impedimentos externos que escapen a la autoridad del Scrum Team, sin convertirse en un gerente de proyecto tradicional?

Respuesta: el Scrum Master remueve impedimentos, protege al equipo de interferencias externas y facilita el proceso Scrum, pero su autoridad llega solo hasta donde llega el equipo. Cuando el problema es estructural, como la falta de recursos o de apoyo de la alta dirección, la causa está fuera de su alcance, porque no tiene poder de decisión sobre presupuestos ni contrataciones. En esos casos, su papel pasa de "resolver" a "visibilizar": puede usar la información que sale del Sprint Review o de la Retrospectiva para mostrarle a la dirección, con datos concretos, cómo esa falta de apoyo está afectando el proyecto. Así influye a nivel organizacional sin convertirse en un gerente tradicional, porque sigue sin repartir tareas ni tomar decisiones de presupuesto: solo comunica, insiste y hace visible el problema ante quien sí puede resolverlo.

Conclusión

Las tres preguntas asignadas a nuestro grupo apuntan a una misma idea central: Scrum ofrece estructura y previsibilidad, con historias en un formato definido, Sprints de duración fija y roles claros, pero esa estructura no resuelve por sí sola la complejidad técnica de un sistema, la necesidad de un flujo de trabajo más eficiente ni las limitaciones propias de la organización.

La independencia total de las historias de usuario es un ideal difícil de cumplir en proyectos reales, por lo que el equipo debe aprender a identificar y planificar las dependencias en lugar de ignorarlas. El sistema pull de Lean y los lotes fijos de Scrum resuelven necesidades distintas, previsibilidad frente a flujo continuo, y Kanban permite combinar ambos enfoques. Finalmente, el Scrum Master tiene un límite claro frente a los problemas estructurales de la organización: puede visibilizarlos y comunicarlos con evidencia, pero no puede resolverlos solo.

Como equipo, esta actividad nos deja claro que Scrum funciona bien cuando se apoya en transparencia, comunicación constante y respaldo real de la organización, algo que consideramos un buen punto de partida para nuestra participación en el foro de debate.

Reflexión individual

Diego Arroyo

Lo que más me llamó la atención fue el tema de INVEST. Siempre había pensado que una buena historia de usuario solo dependía de que estuviera bien redactada, pero ahora entiendo que la independencia entre historias es igual de importante, porque si no se cumple, toda la estimación del equipo se vuelve poco confiable.

Isalleth Sánchez

A mí me quedó más claro el tema de Lean y Scrum. Antes pensaba que Kanban y Scrum eran metodologías completamente separadas que competían entre sí, pero investigando entendí que se pueden combinar, y que Kanban ayuda a que el trabajo dentro del Sprint fluya mejor sin romper la estructura de Scrum.

Einar Delgado

Para mí lo más interesante fue analizar el límite del Scrum Master. Uno tiende a pensar que ese rol puede resolver cualquier problema del equipo, pero al leer sobre los factores de fracaso relacionados con recursos y apoyo de la dirección, me di cuenta de que hay cosas que simplemente no están en sus manos, y que su verdadero poder está en comunicar el problema, no en resolverlo solo.

Justin Barrios

Lo que rescato de este trabajo es que Scrum no es tan rígido como parece a primera vista. El Sprint sí es fijo y no acepta cambios una vez que empieza, pero herramientas como Kanban permiten que el equipo se adapte sin romper esa regla, lo cual me pareció una forma inteligente de mantener el equilibrio entre orden y flexibilidad.

Jorge Cáceres

El ejercicio de pensar en dependencias entre historias de usuario con un ejemplo real, como el carrito de compras y el pago, me ayudó a entender mejor por qué la estimación falla cuando no se detectan esas dependencias a tiempo. Es un problema que probablemente vamos a enfrentar cuando trabajemos en proyectos reales.

Estiven Quintero

Me quedó claro que el Scrum Master no es un jefe ni un gerente tradicional, sino más bien un facilitador que necesita comunicarse bien hacia arriba y hacia abajo. Antes de este análisis no distinguía tan bien la diferencia entre remover un impedimento y simplemente visibilizarlo ante la dirección.

Ahmed Alonso

En lo personal, este trabajo me hizo ver que Scrum, Lean y Kanban no son enfoques opuestos, sino piezas que se complementan según lo que necesite el proyecto en cada momento. Creo que eso nos va a servir mucho cuando tengamos que decidir cómo organizar nuestro propio proyecto integrador.

Bibliografía

- Torres Pila, J. M. (2025). *Dirección de proyectos de Software*. Editorial Deming.