Alcance

* Product Backlog: requerimientos funcionales y no funcionales + gui mockups.
* Project Backlog: actividades de gestión (análisis, planificación, control, mitigación de riesgos), de preparación del ambiente, construcción de tests, deploys, etc.

Herramienta: targetprocess.com (ver como usar stories como actividades de gestion)  
  
WBS: <https://docs.google.com/spreadsheets/d/1EMNyZoh0F8mYEVTnA1jz1ivyoWAEXuWh8LpYCKWTacQ/edit>



Gestión de cambios

* Análisis de impacto: para un cambio propuesto viable se describe cómo afectaría este exactamente al alcance y tiempo (story de la funcionalidad ya planificada que resulta afectada, nuevo riesgo introducido, tiempo insumido, funcionalidades a cancelar, etc) para que el cliente sea consciente de las consecuencias de aprobar e implementar dicho cambio.

Herramienta: usar un template tipo formulario.

Tiempos y costos

* Estimaciones para cada actividad/story:

Unidad: horas hombre

Método: planning poker

* Seguimiento y control:

Análisis de desvío respecto de estimación:

usar un burndown chart sencillo.

Métricas:

cantidad de bugs abiertos vs resueltos

tiempo estimado vs demandado realmente

alcance planificado vs completado

* Sprints de 2 semanas: 20 hh por sprint por integrante

Calidad

* De proceso: usamos Kanban para nuestro workflow y visualizar el estado del mismo. Hay que describir cuál es el criterio para cambiar de columna la story/actividad.
* De producto:
  + Criterios de aceptación para cada funcionalidad (con esto además especificamos los requerimientos no funcionales asociados).
  + Valores aceptables para las métricas identificadas.
  + Plan de pruebas:
    - Qué vamos a probar?: una prueba unitaria para cada story implementada (mínimo).
    - Criterios de aprobación: acordar con el cliente (ej: totalidad de funcionalidad planificada implementada con X cantidad de bugs abiertos).
    - Clasificación de bugs: entre críticos y no críticos, estéticos, etc.

Riesgos

* Identificación: usamos un cuestionario basado en una taxonomía.
* Evaluación: matriz de riesgos bien simple. Columnas sugeridas:

#,

Descripción,

Sprints afectados,

Probabilidad,

Impacto,

Exposición,

Plan de respuesta (hh de mitigación si hay),

Umbral,

Contingencia (hh estimadas si ocurre el riesgo),

Responsable

(solo proponemos un plan de respuesta para los que tienen mayor Exposición)

Expectativas

* Principales interesados: identificarlos
* Discutir objetivos de cada iteración (sprint) junto con el cliente para definir un **Criterio de Éxito** (esto queda documentado en la minuta de reunión).

Comunicación

* Comunicación externa:
  + Informes de avance: alcance completado (estado del backlog), riesgos de interés para el cliente, estado de los cambios solicitados, etc.
  + Reuniones con el cliente
  + Minutas: destacando puntos tratados, acuerdos y compromisos.
* Comunicación interna:
  + **Slack** chat como medio principal
  + Reuniones de planificación semanales
  + **E-mails**

Documentación

* Técnica:
  + diagrama de alto nivel de la arquitectura
  + servicios de la API con ejemplos de ejecución.
* De usuario:
  + descripción del uso para cada funcionalidad implementada (manual de usuario).

Trazabilidad

* Matriz de trazabilidad de requerimientos: para cada requerimiento se lleva un registro del estado del mismo y de los documentos asociados (feature y stories/actividades asociadas, mockups de ui, casos de prueba y resultados de ejecución, bugs asociados, aceptación del cliente, etc)

Retrospectivas

Idea: al finalizar cada sprint se junta el equipo y se deja asentada una “minuta de retrospectiva” que contendrá:

* Keep, Fix, Try
* Lecciones aprendidas