**Chronos Development Process**

**Summary Checklist**

Starting with the prioritized capability list from stakeholder capabilities, and assuming the system architecture is defined, each iteration is tracked on the online Kanban board[[1]](#footnote-1)

https://trello.com/b/oso5IQP2/chronos.

**Release Plan***.* Before any iterations get started (Iteration 0), the Product Owner (PO) identifies the full expected functionality (features)[[2]](#footnote-2) for the product(s). All use cases for Adventurer and Quest Master are listed by theme, roughly equivalent to one or two iterations. Functional dependencies are included. For example, Create a New Hero must be executed before Explore the Castle can be executed.

**The following steps are repeated each iteration.**

1. **Backlog.** The Product Owner (PO) places each use case in the Kanban Backlog area for one iteration. Ideally, all the use cases the PO selects can be done in one iteration. Size each feature in the list to no greater than 13 points each although little is known about each one yet.
2. *Release Plan.* Build the Release Plan by distributing features across all iterations based on the team’s velocity. Historically, that averaged 14 points per iteration.
3. *Iteration 0*. Set aside one iteration period to install the infrastructure, catch up on CYOs, and distribute the info needed for the sprints.

**KANBAN BOARD**

1. The Product Owner (PO) selects the top few use cases for Iterations 1 and 2 from the Release Plan, and places the most urgent ones into the TODO lane (max of seven currently).
2. The team sizes each of the TODO cards and each person selects what they want to work on, using the stage gates (Analysis, Design & Scripting, and Construction & Testing team reviews) and their current task allocations as a guide. Each person works on no more than two cards at a time—a primary and a backup in case their primary is stalled.
3. Each day the team walks the Board from right (DONE) to left (TODO) to move their card toward completion.
4. The team holds reviews of stage gate *packages* for approval. No card is allowed to move pass the team review lane without Greg’ signature. Team reviews are held on the following packages when the constituent artifacts are completed:
   * Analysis package (per use case): Detailed use case, wireframe(s), domain sequence and class diagrams, and GUI test scripts.
   * Design package (per use case): Design-level sequence and class diagrams, integration test scripts.
   * Build package: code and unit testing completed, GUI and integration testing results report, regression testing passed 100%.
5. The iteration coach adds each point from cards that reach the DONE column to the Iteration burn-up chart. The burn-up chart visually shows actual progress against planned progress, scope change, and is easily converted to Earned Value.
6. As cards as removed from the TODO lane, the PO moves more cards into it. The team sizes each card as it appears.

**SPECIAL CASES**

* *High priority items*: If an urgent task comes to light, it is marked as high-priority and someone agrees to take it; their task is then marked as Blocked until they can get back to it. Normally, the PO assigns urgency to a card, but anyone on the team can do that with the PO’s signature if necessary.
* *Impediments*: Some factor that causes a card to stop flowing. Impediments cause the card to be marked as Blocked. Impediments must be escalated after 24 hours as a risk to the iteration release.
* *Blockers:* Cards that have stopped because of an impediment or higher priority assignment, such as the PO pulling a team member off-task.
* *Defects and points:* Defects are also features but without points, and slows down the team’s velocity. A use case is not completed until is has no defects, and no points can be tallied until is it completed.
* *Swarming.* When an unblocked card is stopped for more than 2 days, then the assignee requires some help. The entire team drops what they are doing to help the assignee, and move the card out of its lane. This team action is called swarming.
* *Carry-Overs.* If a task is not completed by the end of the iteration, it is placed back in the backlog or TODO lane for the next iteration. No points are tallied for the iteration in which it was not completed.
* *New items*: Sometime a use case or task is discovered. It is added to the backlog to be moved by the PO to the TODO list. Defects are added to the TODO list as they are found.
* *Release Burn-up Chart*:The collected burn-up chart that combines all iteration burn-up charts to show progress vs. plan for the entire iteration.
* *MMR (Marginally Marketable Release)*. Each use case must be done completely—no outstanding defects or missing features. When the set of features are completed and added to the Build, the Build is considered ready for user demo, and possible production release (a business decision of the PO or higher management).

1. See *Kanban: Successful Evolutionary Change for Your Technology Business*, David Anderson, Blue Hole Press, 2010. Anderson is the first person to convert the Toyota assembly line process to software practices. [↑](#footnote-ref-1)
2. Kanban refers to each use case, task, or dependencies—anything that shows on the Board—as a feature, including defects. For clarity, this document will refer to “cards” instead of the word feature to avoid confusion. [↑](#footnote-ref-2)