Avalonization

Moving to Avalon lifecycle interfaces

This page is for documenting the current plan for moving to the Avalon lifecycle interfaces. This page should be a summary only. If you agree/disagree with a particular point, please cast you vote on this page with your name.

- Drop the stratum lifecycle interfaces. They will be replaced with the Avalon lifecycle interfaces. This will be done for Turbine and Fulcrum.

+1 QuintonMcCombs

- The services layer and all component composition/management code will be removed from Fulcrum. This will leave Fulcrum as a collection of components only.

+1 QuintonMcCombs

- The coupled services of Turbine will remain unchanged. They should be thought of as deprecated in favor of creating new services as Avalon components. We will need to maintain this code for bug fixes but focus enhancements on the Fulcrum components.

- We need a logging solution that will take advantage of atleast log4j if not commons-logging. We should try to keep commons-logging to allow user to pick their actual logging solution. This solution should be a component that we could pass to our components.
  Here are some ideas to think about.
    - The development version of ECM will allow log4j use. Perhaps Fortress or Merlin will allow for it as well
    - We could write a custom Logger that will wrap commons-logging.
    - We could borrow some code from Plexus for logging.
- This has been completed. We are using development versions of the framework and ECM. Log4J logging is supported.

- On the fly conversion of the existing properties format into an Avalon Configuration object.
    - This would only be a short term solution. Users would be expected to move to the new configuration format at some point.
    - We could also simply change Fulcrum to require the XML configuration. Since Fulcrum has never had a release, we could do this....

Either way - +1 QuintonMcCombs

- Could maybe be done with some sort of CompositeConfiguration? Merge the old format and the format together? EricPugh

---

- Components in Fulcrum need to be compatible with Plexus, ECM, Phoenix, Merlin, and Fortress.
    - This is really more of a very nice to have than an absolute requirement.
    - Mainly, we want to use the Servicable interface over Composable.

+1 QuintonMcCombs

- We should try to release Fulcrum at the same time as Turbine 2.3 if possible.
    - Does this mean that we also need to work on catching the Fulcrum components up with the changed to the services coupled with Turbine?
    - Do we want to use other component from Fulcrum by default in the T2.3 release?

Releasing at the same time - +1 QuintonMcCombs

- We need a container for Turbine 2.3.
    - ECM?
    - ECM - dev version that allows Log4J logging?
    - Merlin?
    - Fortress?
- ECM is being used for now.