Architecture Update Design

Driving idea:

An AU is defined as it's own branch of the arch-as-code workspace that contains a yaml documenting that AU, and has updates to the model.

Key:

```
[O]   = provided as a arch-as-code command line argument, or from something like a ~/.arch-as-code/credentials.json file
[M]   = must be entered manually by editing the yaml
[P1]  = obtained from P1 automatically
[A]   = automatically generated programmatically
[VCS] = not captured in yaml at all-- information comes for free from the version control that the arch-as-code repo will be under (such as the name of a branch or commit, etc.)
...   = array
?     = more analysis required
```

Information that goes in the Architecture (model yaml)

```
model updates [M]
```

Information that would be contained via VCS

```
arch.jira [VCS] (would be in the PR title/description, or the branch name)
model updates' descriptions (would be in the commit history, PR description, etc.)
```

Architecture Update Yaml Format:

```
name [O]
identifier [A]
milestone [P1]
jira epic ->
    ticket [A] (when feature stories are created)
    link [A] (when feature stories are created)
authors ->
    author... ->
        name [O]
        email [O]
PCAs ->
    PCA... ->
        name [M] (Future: [A] sourced from a "product master" google sheet)
        email [M] (Future: [A] sourced from a "product master" google sheet)
P2 ->
    link [P1]
    jira ->
        ticket [P1]
        link [P1]
P1 ->
    link [P1]
    jira ->
        ticket [P1]
        link [P1]
    summary [P1]
```

```
useful-links -> (might not be necessary to duplicate from the P1)
    link... ->
        description [P1] / [M]
        link [P1] / [M]
milestone-dependencies -> (might not be necessary to duplicate from the P1)
    dependency... ->
        description [P1]
        links ->
            link... [P1]
decisions ->
    decision... ->
        type (ITD, IFD, or SSD) [P1]
        info ??? [P1]
        tdds... ->
            tdd-id / tdd-alias [M]
ACs ->
    AC... ->
        id [P1] (Special case: [M] if new AC only in AU)
        text [P1] (Special case: [M] if new AC only in AU)
        tdds... ->
            tdd-id / tdd-alias [M]
TDDs... ->
    Component affected ->
        component-id / component-alias [M]
        TDD... ->
            id / alias [M]
            text [M]
```

Things to validate:

- All decisions from P1 are covered by >=1 TDD.
- All ACs from P1 (or ACs that were added manually) are covered by >=1 TDD.
- All TDDs refer to valid components in the model.
- There are no TDDs that are not referred to by an AC or decision.

How to visualize:

- Some way to visualize the model in each branch is necessary
- Diffs not necessary for now