Ticket application - epsilon version
==

Sample Grails application meant as a teaching tool. As always, the code in the integration and units tests is as instructive (is not more) than the __real__ application code.

Walkthrough
----

1. Start by creating 3 domain classes. A `Ticket` has a summary and a description. A `Developer` is assigned many tasks that she is meant to complete. Many `Tasks` can be necessary to close a `Ticket`. Many `Developers` can work on the same `Task`. Each `Developer` has a full name and a work email. Finally, the creation date of `Tickets` and `Tasks` must be recorded.

```
$ grails create-domain-class ca.mcgill.epsilon.Ticket
$ grails create-domain-class ca.mcgill.epsilon.Task
$ grails create-domain-class ca.mcgill.epsilon.Developer
```

1. Create integration tests for the new domain classes and verify that it is possible to save new instances, to wire them together and to find them in the database. While you're at it, check that the auto-timestamp feature described above actually works.

```
$ grails create-integration-test ca.mcgill.epsilon.Ticket
$ grails create-integration-test ca.mcgill.epsilon.Task
$ grails create-integration-test ca.mcgill.epsilon.Developer
...
$ grails run-app integration:
```

1. Add validation to summary, description, fullName, workEmail. Update your integrations tests.

1. A `Ticket` has a type like `Bug`, `Improvement`, `New feature` or `Question`. A `Ticket` also has a priority that ranges from 1 to 5. Dont forget to update your tests.

```
$ grails create-domain-class ca.mcgill.epsilon.TicketType
```

1. A `Ticket` has a particular life cycle. It starts as `pending`, it is then `assigned` to a developer by a manager. When the developer starts working, it is `in progress` and when she finishes it becomes `closed - fix`. It can also `closed - wont fix`. For statistical purposes, we also need to keep each status change. Furthermore, each status change must be sorted by timestamp. Finally (!), a `Ticket`, when saved, should be `pending`. Dont forget to update to update your tests...

```
$ grails create-domain-class ca.mcgill.epsilon.TicketStatus
$ grails create-domain-class ca.mcgill.epsilon.TicketStatusChange
```

1. A `Ticket`can have one file `Attachment`. It's important to save not only the file content but also the content type (application/pdf, image/png, etc) and the original file name.

```
$ grails create-domain-class ca.mcgill.epsilon.Attachment
$ grails create-integration-test ca.mcgill.epsilon.Attachment
```