minimum viable product

Daniel Jackson
why design matters

sample costs* (in person-hours)
› purpose & goals (3)
› context diagram (1)
› object model (4)
› wireframes (2)
› discussions (10)
› code (300)

* for Area 2 system
› tracks graduate student progress in CS
cost of defects

chart from https://buildsecurityin.us-cert.gov/articles/best-practices/code-analysis/business-case
V model

used by large contractors (eg, Accenture)
› outline acceptance tests during requirements

when do you discover you got the requirements wrong?

diagram from http://www.eccam.com/ibm_requirements_management.html
fail early!

think before coding
› work hard on problem analysis & design
› use precise notations (eg, object models)
› run automated analyses

reduce time to deployment
› to check if problem exists: AdWords, landing pages
› to check usability: paper prototypes
› to check concepts & features: deploy MVP
the learning loop

Minimize *TOTAL* time through the loop

but you have to learn!

not necessarily quantitative

**MVP**

minimal viable product (MVP)
› or “Simplest Thing That Works” (STTW)

designed for learning
› is this a product people want?
› will people use it in the way we expect?
› are their hidden gotchas?

how to do it
› write a list of feature descriptions
› take an axe to it: see how many you can remove
› look at data model, and cut out all but essential features
› ... and then try and reduce it even more :-)

what not to do
› build a shell with standard features (eg, login) & nothing distinctive
› a useful test: is at least one key concept represented in the MVP?