

Initial Advice & Tech Plan For A Hypothetical Web Startup

by Mike Kramlich

2016 June 9

identify biz leadership's desired goal state & changes in terms of:

- features, quality, stakeholder milestones

- for launch, launch +1 month, +3mos, +6mos, +1year, exit/horizon

- becomes your roadmap

- ok if not know all this, ok if not want to be too concrete about future

- ok if changes, helpful to know now, healthy exercise

- ok if soft launched (beta) already and if the plan is for a gradual incremental ramp-up of beta members; and growth after "launch", never in a steady state

current state discovery:

- UIs, code, arch, data model/size, algos, databases, queues, caches, servers, integrations, dependencies, tech, langs, libs, frameworks, tools, process, docs

- completeness, quality, relevance of each above

- beta site: users, traffic, latency, hosting, resource usage (cpu, mem, disk, network), errors (via logs, alerts, UX)

determine if recommend to add to, optimize, fix, remove, replace/rewrite, or leave alone, any or all elements above

get buy-in, then do it

after ensuring all desired launch features ready, next pre-launch priority is make arch/code/data changes to reasonably ensure that site will both launch & continue to be:

- fast, scalable, have fresh raw data, correct conclusions for customers, minimal/zero glitches, gracefully fail, have redundancy, automation, minimal operating cost, desired biz visibility

build or buy or rent or download:

- decide for every element

- bias to build only the most biz unique/IP-worthy/differentiator/value-add, bias to less cost, less burden on staff, most agility, most visibility, most prod-proven, less reinventing wheel

- bias to FOSS, SaaS, PaaS, IaaS, cloud, VM's, Linux, CLI's, web, mobile

- eg. monitoring should be mostly a rentable SaaS with a touch of in-house glue code

prototyping & AB testing

- try to make fast/easier/cheaper for biz to do, if not already have way

tech documentation & arch diagrams

version control (brief thumbnail: git)

code review

testing

releases

security

monitoring

routine site support

disaster mitigation

backups & restores

team:

skills, roles, workload, overlap, fallback, redundancy, avail, fit, culture, cost

with at least *some* pass at all of the above in place (however simple at first, however fast, temporary or cheap, because its a judgment call, and all should be tailored based on situation, time pressure, resources, budget), next do:

launch plan

launch

watch, analyze, iterate, revise, revisit, reconfirm, tune, evolve, improvise, repeat, grow biz, expand team and upgrade the tech over time

always be learning. adapt. prioritize

have fun!