**4/17/2017**

--monitoring and deployment

--quite integrated now b/c of continuous deployment

--server health/service health

--1st level of monitoring: heartbeat

--queue length

--2nd level of monitoring: load factor: automatic systems spin-up/down as needed

--cost factor

--need monitor displays: 2-D grid with different colors or something

--avg. proc. Time (requests/sec.)

--alerts: opposite of monitoring. Text message is good.

--alerts need logs and such -> different alert messages for different problems

--how is my requests/sec. changing over time?

--need long-term monitoring

--Netflix simian army

**4/24/2017**

--PieMatrix as microservices: deployment monitoring plans—long and short term metrics—how to get from test implementation to where the user sees changes in real implementation.

--think about whether you trust developers or need QA.

--what warnings/errors and how many are ignored? For time reasons…

--updating with liveness: what to do?

--using correct versions of external libraries with current PieMatrix version.

--upgrade CSS style, for example, for all pages.

--remote libraries: what if they disappear?

--password management? What if server crashes and restarts? Where’s the friggin’ password?

**5/1/2017**

--dev -> SCC

--trigger automated testing?

--qa get a shot at it?

--code reviewers?

--deployment tools: Octopus -> .net focused

--deployment -> production?

--deployment -> staging?

--deployment -> customer acceptance and testing (cat)

--dev->qa->staging->cat->production

--rolling-live-updates to maintenance windows

--how healthy are services? Heartbeat, throughput, queue length, etc.

--what makes the most sense?

--IBM sells super-computers by the hour

--long-term monitoring: t