The CAP Theorem of Humans

An exploratory analysis of computing theory, management theory, and economic theory, in an effort to build resilient and effective teams.

Christian Witts

DevOpsDays Cape Town 2017

(pretty much everywhere) @christianwitts

Who am I ?

- Manager for 6 months
- Software Engineer for 13 years
- Writing code for 29 years
- Anti-social socialite
- Soon to be a husband
- Aquarian, who likes long walks on the beach
- Enjoys metal far more than EDM
- Not sure if he’s cut out for management

Agenda

- A brief recap of CAP theorem
- How are humans different from computers
- Methods to aid in system stability
- The levels of leadership
- The 4 steps to inventiveness
- Lessons learnt
- Final thoughts

The CAP Theorem: A recap

- Consistency
  - Every read receives the most recent write, or an error

- Availability
  - Every request receives a (non-error) response - without guarantee that it contains the most recent write

- Partition Tolerance
  - The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes

In other words, the CAP theorem states that in the presence of a network partition, one has to choose between consistency and availability.

How humans differ

- Consistency
  - Humans are generally only consistent when executing the exact same task, which is a rarity outside of day-to-day work

- Availability
  - Humans are available, only when waiting for work, and even then not 100% of the time as you spend time in ideation and creative solution thinking modes

- Partition Tolerance
  - Humans can survive partition tolerance, due to the free-thinking nature of humankind, although consistency tends to take a knock unless guidelines are in place

In other words, humans are about as reliable as UDP; ie. mostly reliable, but don’t ACK requests.

cont.

Think of your organisation as a distributed monolith and microservice architecture that needs delicate orchestration. Ensure you facilitate a modicum of structure regarding your communication protocols, this includes, but not limited to

- Well defined methods of communicating, as well as SLAs on response time
- Knowing when to break the circuit, isolate the work
- Limit the rate of work so people aren't on 24/7

No one wants to be burnt out, and a failure to react to circumstances within your reach to change, can lead to critical failures in your team.

Consistency

- Everything should be consistently reproducible
    - Configuration/Infrastructure as Code
    - Playbooks / Runbooks
    - Documentation needs to be accurate
    - Single source of truth
    - You want to get the same answer every single time

Availability

- No resources should be blocked for longer than is truly necessary
  - Remember the wait-time graph

- There should be a clear common, well defined, and visible queue
  - Use the tools available, for example Jira, Clubhouse, Kanban boards, or GitHub issues
  - No one should have authority to override the queue, except in the event of a P1 outage

Partition Tolerance

- Your outcomes should still be possible, even during a split brain scenario
- If half your organisation is on a plane, and you experience a critical issue, you should have no issues resolving it
- If half your organisation leaves the company, are you still able to operate ?
- The direction of the organisation should be clearly mandated, and everyone should have a fair to great understanding of it

Some tactics to employ

- Radical Candor
- Fail Fast
- Blameless Post-mortems
- Breaking Down of Silos
- Effective Communication
- Essential Training
- Mentorship
- Cross-team Pollination
- Explicit Knowledge Sharing

Radical Candor

- Openness
- Honesty
- Authenticity
- Vulnerability
- Culture of feedback

Fail Fast

- Willingness to experiment
- Knowing when to cull projects
- Start small with hackathons, with an emphasis on delivering

Blameless post-mortems

- Be deliberate about post-mortems

- Develop the organisational muscle around post-mortem fundamentals

- Work as a group at recognizing the human tendency to blame

- Actionable post-mortems

Destruction of silos

- Knowledge is meant to be shared

- Unless there is a justifiable reason, open should be the default policy

Effective communication

- Lines of communication should be clear and well defined

- Comms should be succinct, not needlessly cluttered
    - Prefer timeboxed meetings with set agendas

Essential training

- Training should be provided, and opportunities afforded to the employee to exercise that allowance
- 10% time is only 16 hours a month, and most people are only productive 100 hours of the month

Mentorship

- Everyone should have a mentor
- Not everyone wants to be a mentor, so ensure the mentor-mentee pairing is suitable
- If there are no potential mentors inside the organisation, the mentee should be free to seek guidance from outside sources

Cross-team pollination

- There should be a healthy cross-pollination between various teams

- It should not be limited to only the engineering departments
    - You can learn a lot from people outside your role

- Everyone can learn something. Anyone can teach something.

Explicit knowledge sharing

- Brown-bag sessions, aka lunch-&-learns, are a great method for explicitly sharing knowledge

- No one will attend every session, so if possible, take notes / recordings
  - Using a tool like Zoom is handy for this, as you can record and save your meeting / screencast for later sharing

- Although the sessions should generally be focused on business goals and technology, allow people to share knowledge about subject matter outside of your normal scope

The 5 levels of leadership

- Highly capable individual
  - Makes productive contributions through talent, knowledge, skills, and good work habits
- Contributing team member
  - Contributes to the achievement of group objectives, & works effectively with others
- Competent manager
  - Organises people and resources towards the effective and efficient pursuit of predetermined objectives
- Effective leader
  - Catalyses commitment to, and vigorous pursuit of a clear and compelling vision, and stimulates the group to high performance standards
- Executive
  - Builds enduring greatness through a paradoxical combination of personal humility and professional will

The 4 steps to inventiveness

- Don’t assume things have been done in the best way
- When a new task or change effort emerges, encourage creative ways on getting it done
- Experiment with different ways of organising work, find alternative methods for grouping and linking people
- When working to understand your current environment, ask yourself, what other options are possible

Final thoughts

DevOps promotes a culture of rapid experimentation, which allows us to adjust the frameworks over time, and find the right fit.

No one person has all the answers, and we need to take it upon ourselves to break the mould, and successfully iterate to become the best individuals, team members, and leaders we can be.

“Be like water making its way through cracks. Do not be assertive, but adjust to the object, and you shall find a way around or through it. If nothing within you stays rigid, outward things will disclose themselves.

Empty your mind, be formless. Shapeless, like water. If you put water into a cup, it becomes the cup. You put water into a bottle and it becomes the bottle. You put it in a teapot, it becomes the teapot. Now, water can flow or it can crash. Be water, my friend.”

- Bruce Lee

Questions ?