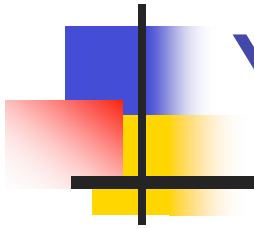
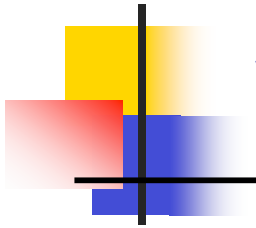


You Scaled Your What?

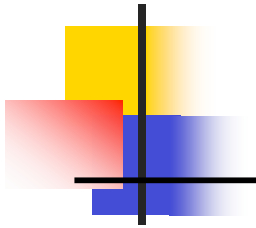
Dan Pritchett

eBay, Inc.



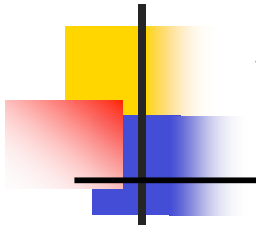
What is Scaling?

- Common definition
  - The ability of a system to accept increasing transaction loads.
- Common focus
  - Processing time per transaction
  - Performance of software and hardware



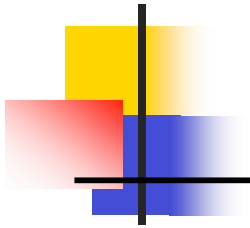
It's Not Enough

- Transactional scaling is but one dimension
  - And it is influenced by most other dimensions
- Scaling applies to all of the "*ilities"



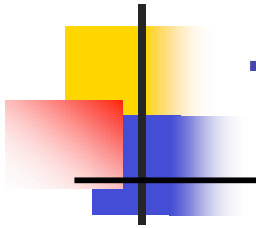
Why Scale All "*illities"?

- An architecture is a complete organism
  - Evolution is a natural part of any architecture
  - Scale is a requirement of any evolutionary path
  - Incomplete scale leads to imbalance
  - Returning to balance can be expensive and time consuming



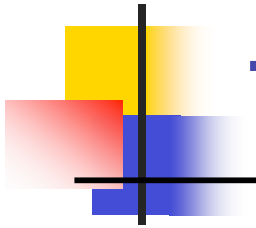
Dimensions of Scaling

- Software
  - Transactional
  - Data
  - Operational
  - Deployment
  - Power
- Process
  - Productivity
  - Time to Market
  - Flexibility



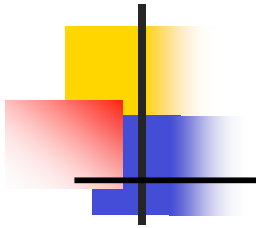
Transactional Scaling

- Probably the best understood
  - Units
    - TPS
    - Response Time
  - Focus
    - Application performance
    - Traditional benchmarking and tuning



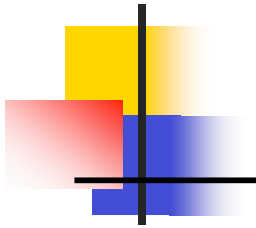
Transactional Headroom

- How long can your business survive?
  - Two critical metrics:
    - Time-to-Live (TTL) on current resources
    - TTL on maximum plausible configuration
- Should be measured regularly
  - Production bottlenecks may appear that were not anticipated
  - Tracking headroom will identify issues before they are a crisis.



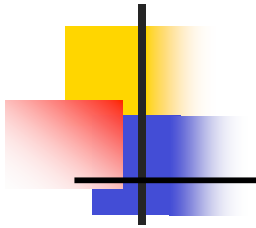
Data Scaling

- How well does your data scale?
  - Units
    - TPS (read and write)
    - Storage
      - ✓ Capacity
      - ✓ Classification - Is all data equally important?



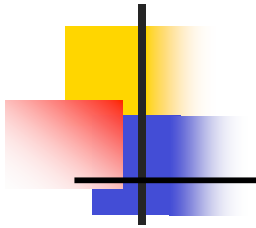
Data Scaling (cont.)

- Functional Decomposition
  - Group data by various qualities:
    - Logical relationship
    - Transactional volumes
    - Business importance
      - ✓ Storage classification
      - ✓ DR requirements
- Transactional Scaling
  - Horizontal scaling
  - Some cases, vertical scaling
    - Try to avoid relying on vertical over horizontal



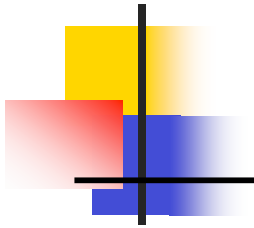
Operational Scaling

- How hard is your software to run?
  - Units
    - Operational staff headcount
    - MTBF
    - MTTR



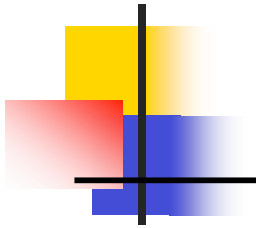
Operational Scaling (cont.)

- Operational scaling is a software problem
  - Complete architectures build in operational concerns at the beginning
- Operations needs
  - Monitoring
    - Status panels, telemetry, alerts
  - Command/Control
    - Update application state live, tune applications live



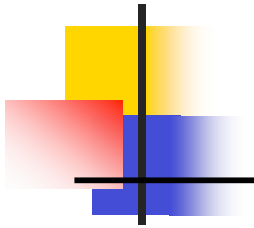
Deployment Scalability

- Can your software support multiple deployments?
  - Units
    - Configurations and topologies
  - Focus
    - Independence from physical manifestations
      - ✓ Hardware
      - ✓ Network topologies
      - ✓ Data center topologies



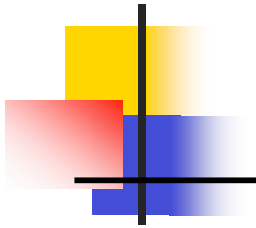
Code Deployment

- Typically an afterthought. Do you design for?
  - Rolling application into production without downtime
  - Rolling **back** from production without downtime
  - Managing component dependencies in a sane way during code deployment
    - Hint: Turning off production isn't sane



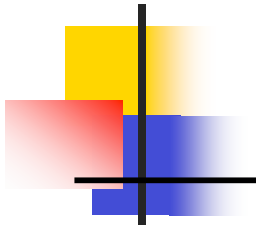
Power Scalability

- Is your application green?
  - Units
    - TPS per watt
- The power crisis
  - Power is quickly becoming the limiting factor in data center design
  - Many large sites will experience transactional scaling limits due to power constraints



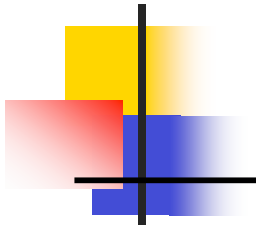
Scaling for Power

- Efficiency
  - Software efficiency returns as important
    - Wasted clock cycles is wasted watts
    - Sanity must still prevail though
  - Stop expecting Moore's Law to fix poor coding decisions
- Utilization
  - Availability requirements often drives down hardware utilization
  - Consider virtualization technology.



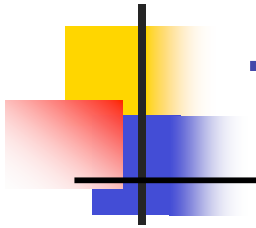
Productivity

- How efficient are your software engineers?
  - Units
    - Treachery lies here
      - ✓ LOC/day?
      - ✓ Features/quarter?
  - Focus
    - Developer throughput
    - Code quality and factoring



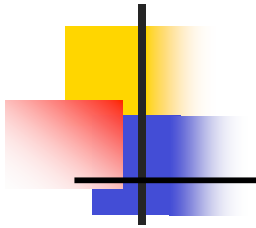
Perils of Productivity

- Metrics for developer productivity are wrought with danger
  - Lines of code
    - Do you want good code or lots of code?
  - Features
    - What is a feature?
      - ✓ A new page layout
      - ✓ An ad server
- What you measure is what you'll get!



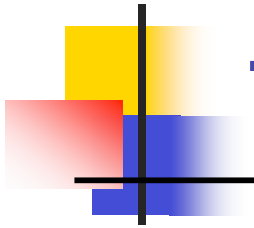
Time to Market

- How quickly can new features be delivered?
  - Units
    - Feature completion time
  - Focus
    - Related to productivity and flexibility



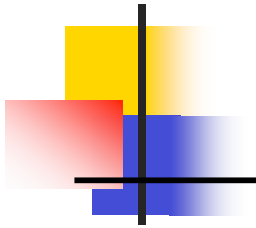
Flexibility

- How easily can your application adapt to new capabilities?
  - Units
    - Development time
    - Code impact
  - Focus
    - Good code factoring



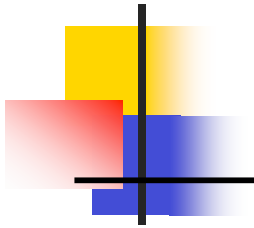
TTM vs. Flexibility

- Feature speed is the antithesis of good code in most organizations
  - Business wants the current feature now
  - Engineering wants clean code for the long term
- Cost of speed inadequately quantified in most organizations



Scalability Realities

- All scalability dimensions are related
  - Each impacts the other
  - Any dimension ignored will evolve into a problem for the application
- Breadth of scalability is routinely ignored
  - Architects become myopic
  - Not factored into business bottom line

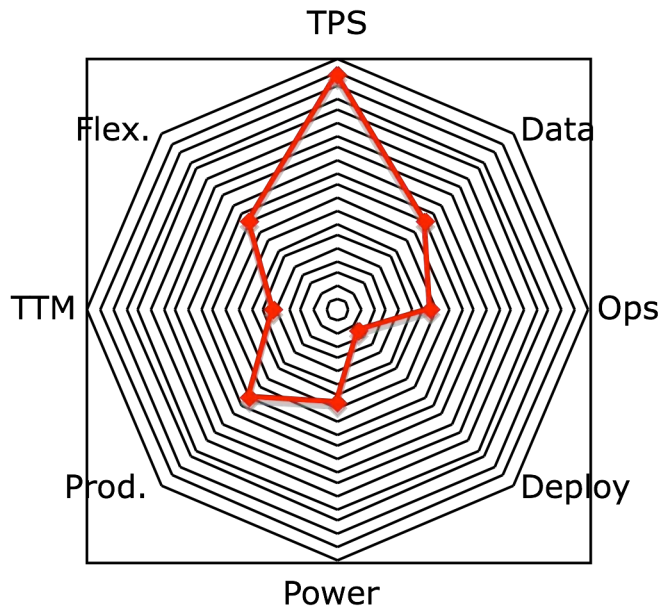


What should you do?

- Make a complete list of the dimensions that matter to your organization
  - This list is a minimum set…you may have more
- Begin measurements
  - First we have to admit we are powerless over our scalability vectors

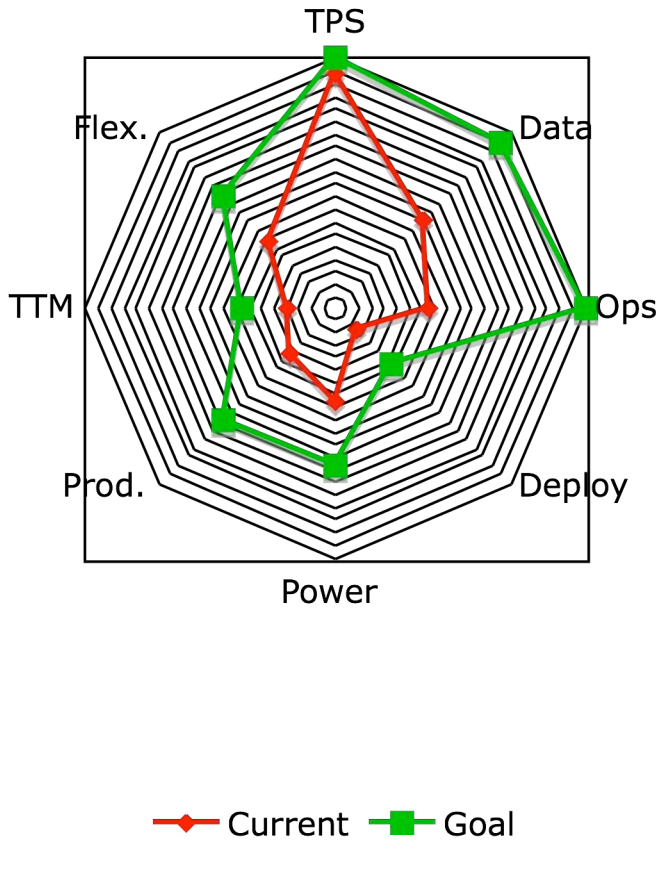
Visualize the Problem

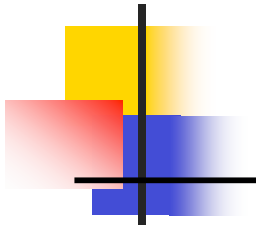
- Radar Chart
  - Challenge
    - Normalizing each metric against ideal value (e.g. what is 100%)
  - Advantage
    - Illustrates skews and problems clearly
- Start with current situation



Define the Goal

- What do you want your chart to look like?
  - Can't reach 100% on all axis
  - What are the correct trade offs for your organization
  - Let chart guide your investments





Summary

- Scalability is broad
  - More than just transactions
  - All dimensions are interrelated
- Measure
  - What you don't measure will be the dimension that becomes the limit to your application's potential.
- Establish organizational goals
  - Architectural and business