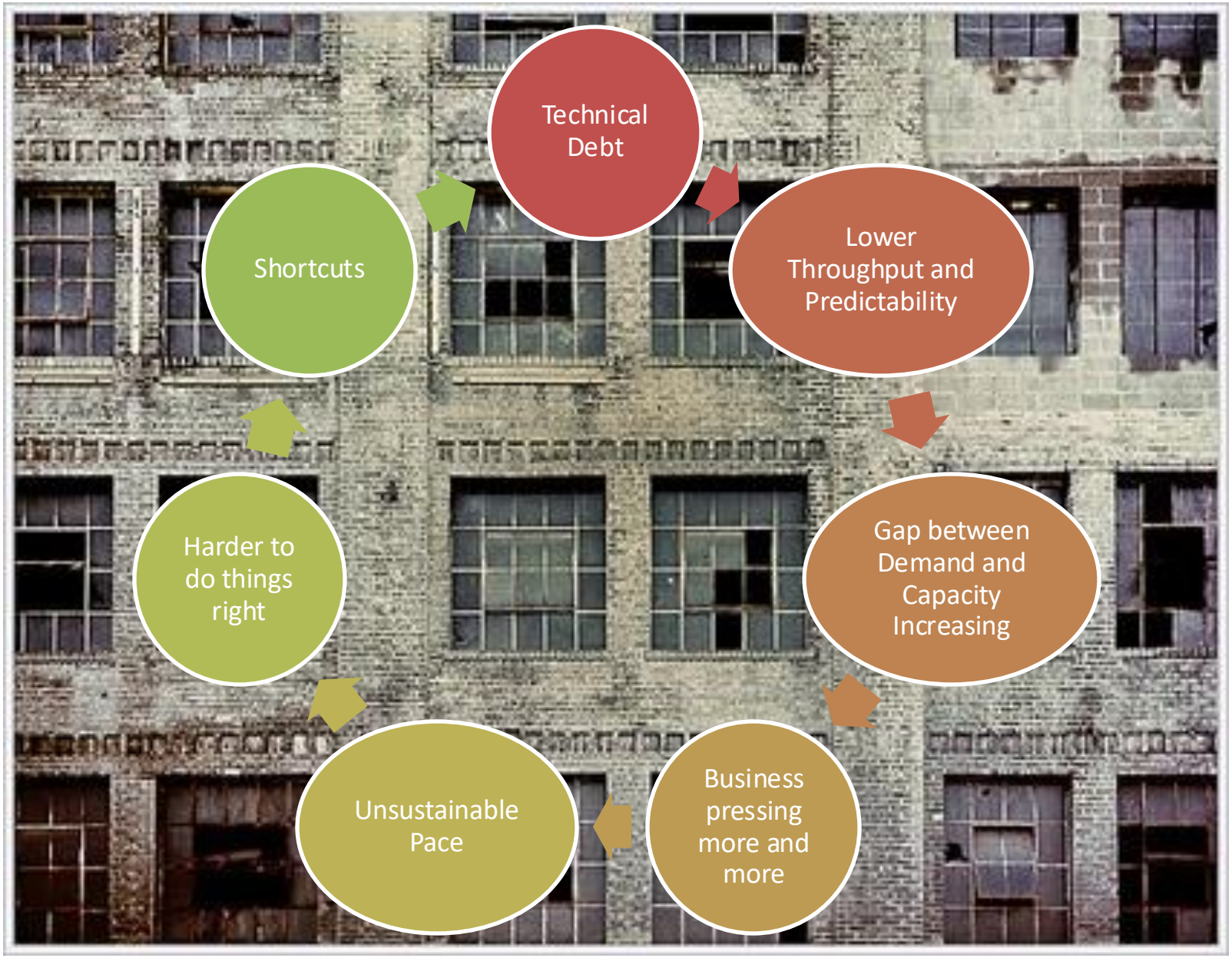
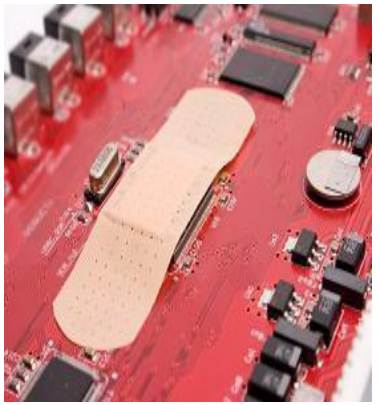
Dealing with Technical debt
Getting to a Sustainable pace

Unsustainable pace / Technical Debt vicious cycle



What are the typical symptoms of Technical Debt / unsustainable pace?

- The Product Dev Factory feels like it cannot succeed
- lower professionalism in engineering
- Mountains of defects
  - High reopen rate
- Mastery of shortcuts and regression impact avoidance
- Many branches and versions that are supported
- No slack for errors
  - Murphy happens
  - Problems are magnified

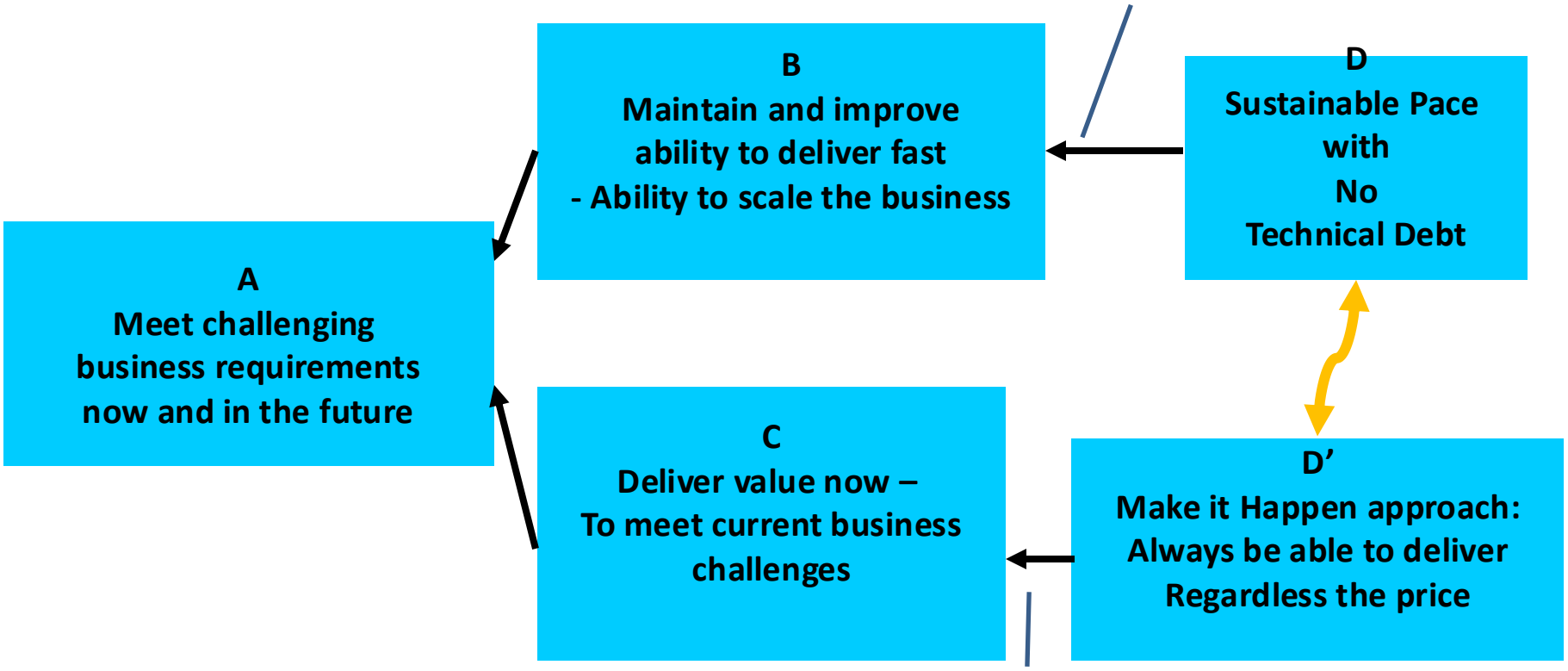


How to get to sustainable pace

1. Measure actual capability and start to work according to it
2. predictability - regain trust
   - In the short term – hard commit on less content for each delivery
   - Soft Commit to deliver as much high quality features as possible
3. Restrain Sales – manage overcommitment
4. Elevate capacity to align with business needs
   - After factory is stable and scalable

Sustainable Pace vs Make it Happen

1. Delivering fast requires good technical infrastructure, robust system
2. Technical debt slows us down significantly
3. Scaling requires simplification of the system, less reliance on heroes



1. With current capabilities, must have make it happen approach otherwise R&D don't deliver enough
2. R&D need to be pushed in order to deliver what the business requires

The solution – differentiated services

Normal

- High Quality
- Low Debt

Expedited

- Value over Quality
- Limited usage

Engineering Improvements

- Build the force
- Return Debt

Small Wins / fast lane

- Deliver frequently
- Customer Success
- Voting by the Business

Technical Debt Backlog

- Identify main areas of debt
- Classify them
  - Cost of debt (Interest...)
    - Frequency of touching the area
    - Amount of pain in touching the area
  - Cost to return debt
- Prioritize
- allocate % of capacity