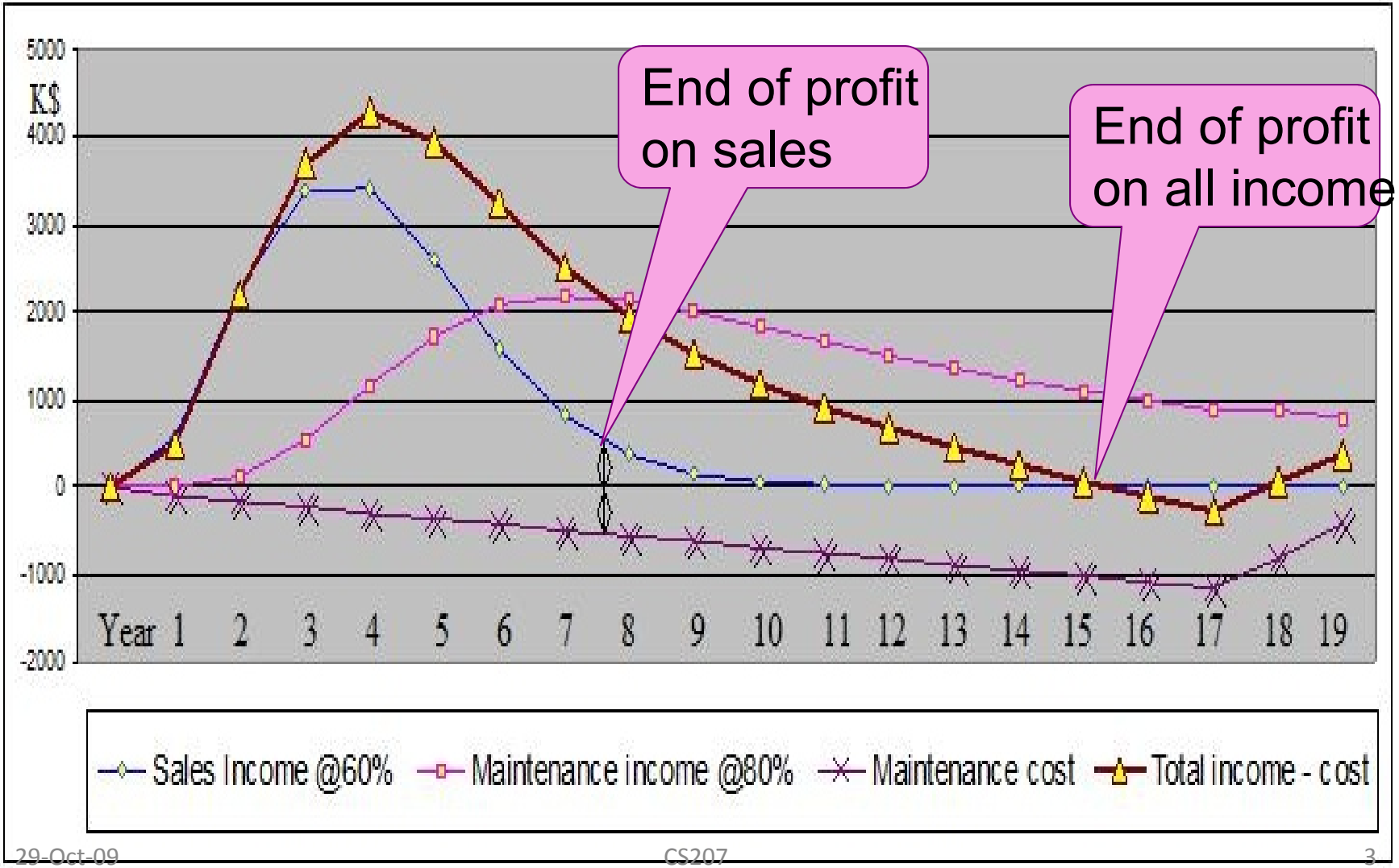
CS207 #5, 23 Oct. 2009

Gio Wiederhold
Gates B12
Homepage at
https://cs.stanford.edu/wiki/cs207/Main/HomePage

Syllabus:

1. Why should software be valued?
2. Open source software. Scope. Theory and reality
3. Intellectual capital and property (IP).
4. Principles of valuation. Cost versus value.
5. Market value of software companies.
6. Sales expectations and discounting.
7. Alternate business models.
8. Allocation
9. Life and lag of software innovation.
10. The role of patents, copyrights, and trade secrets.
11. Licensing.
12. Separation of use rights from the property itself.
13. Risks when outsourcing and offshoring development.
14. Effects of using taxhavens to house IP.

Net income, after sales cost



Software users & IP

Companies that

1. develop & sell software: Topic up to now
   - Basis of IP: income from sales
2. purchase & license software for internal use
   - Do not generate IP with software
3. develop software internally for their own use
   - Basis of IP: relative SW expense × all income
4. combinations

Allocaction

- When there are multiple products
- When there are other contributors to income
  - Substantial hardware
  - Financial consultants in financial firms
  - Experts in call centers
  - Brand name

  Not all of the income can be allocated to the software
- Pareto Optimum

Pareto Optimality

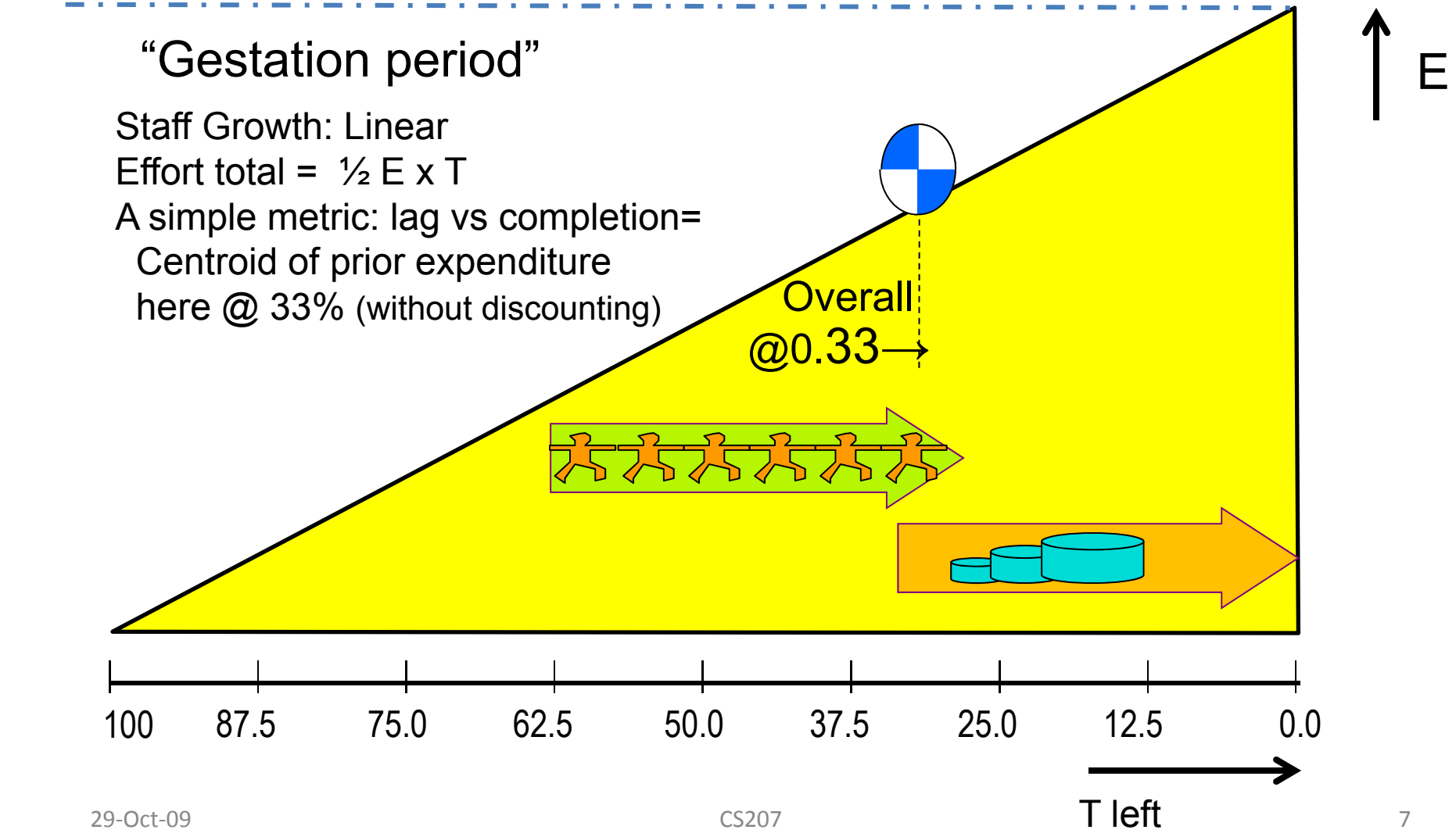
The point were any change lowers the total

- Spending more on software will have less benefit than spending on other stuff
  - People
  - Hardware
  - Advertising
    - For large 10 IT companies the average value allocated to their brand name is 22% (BW survey).

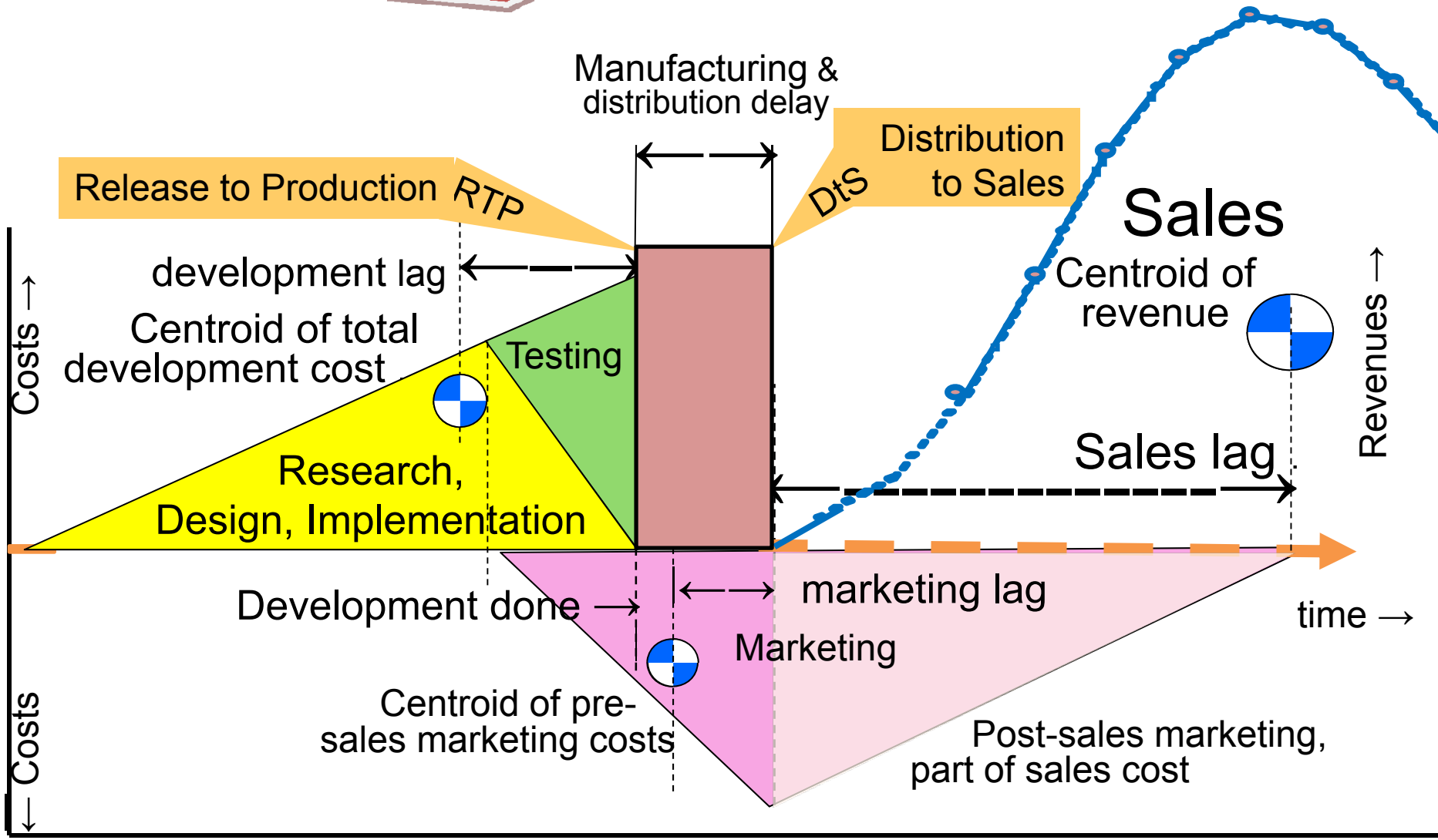
Conclusion:

- If a company is managed optimally, we can allocate IP contribution by multi-year spending patterns

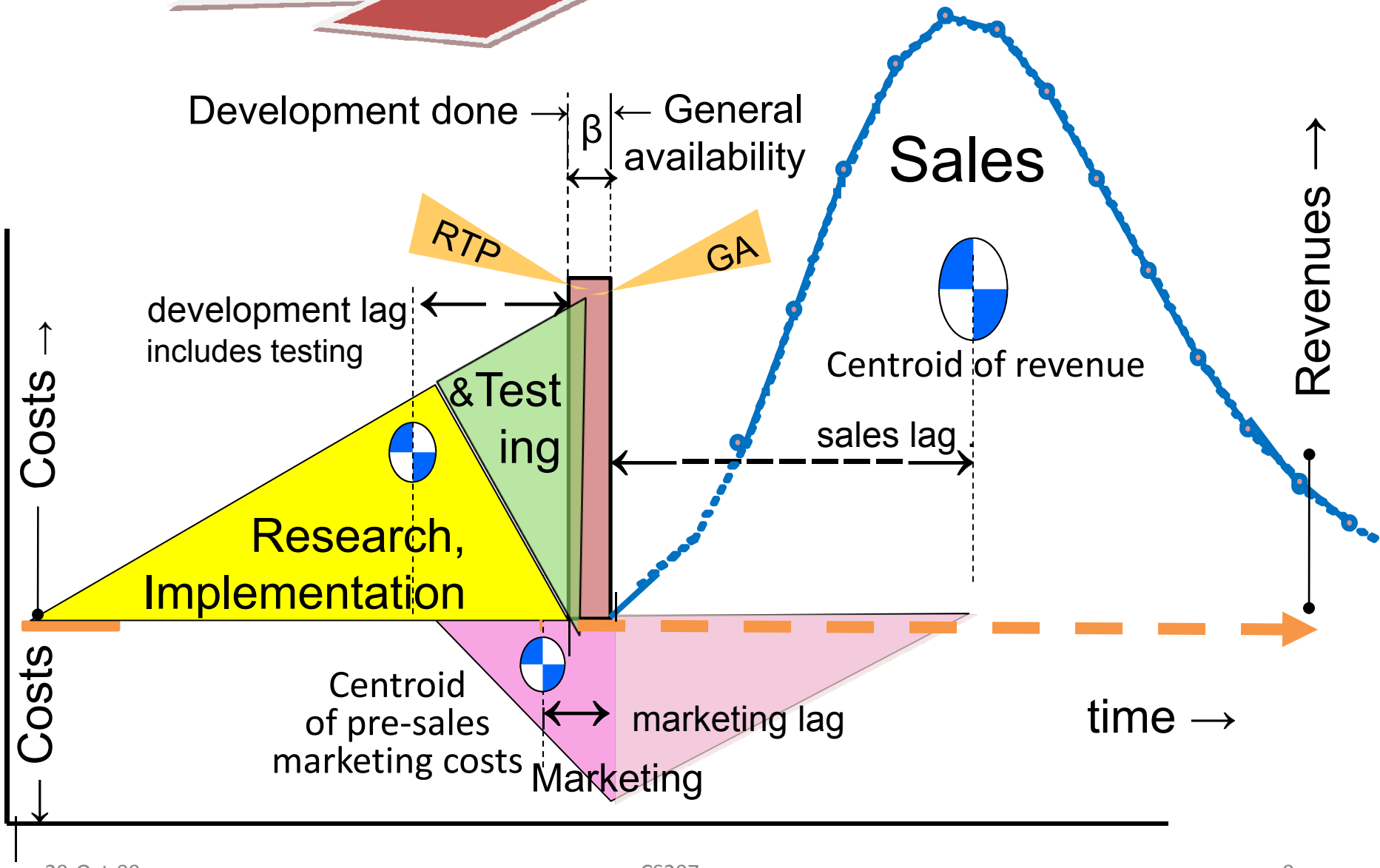
Lag measures
Prior effort



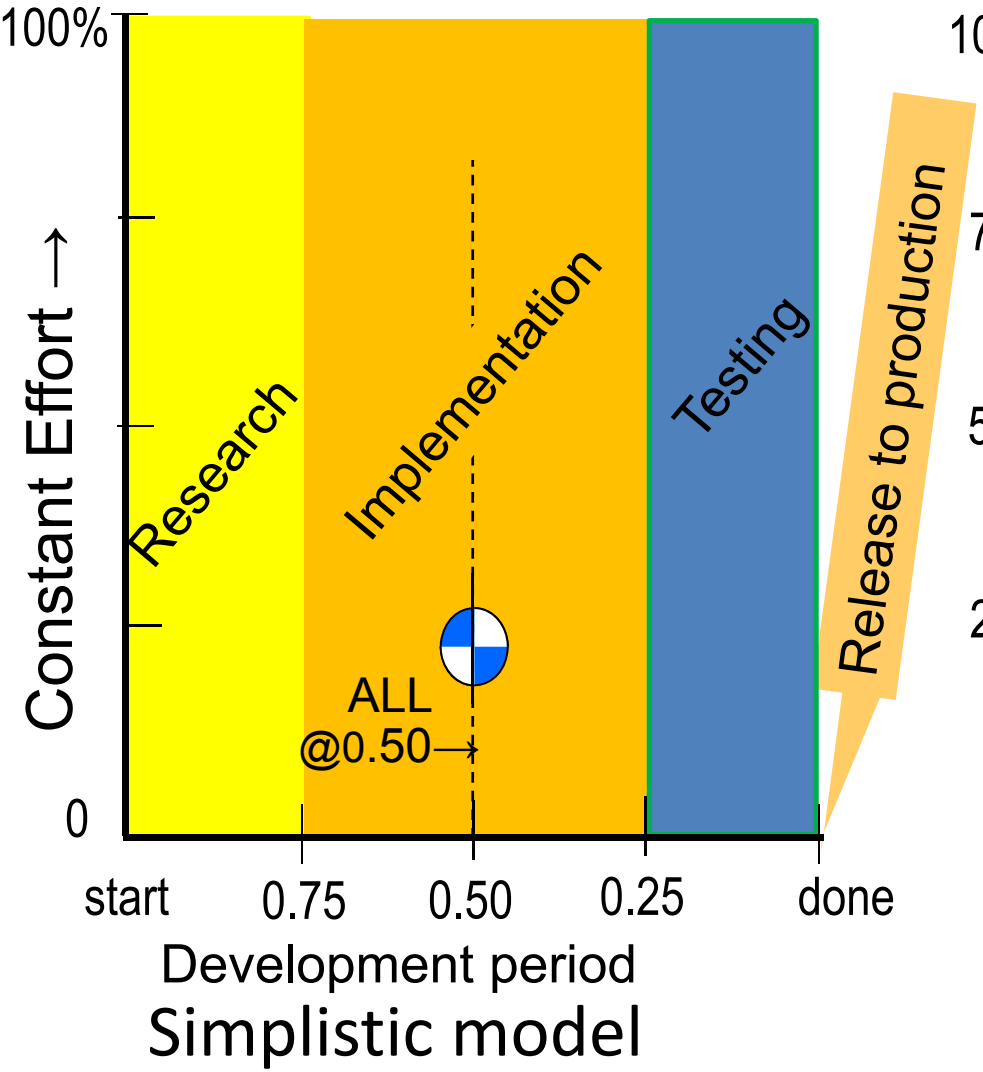
Various Lags



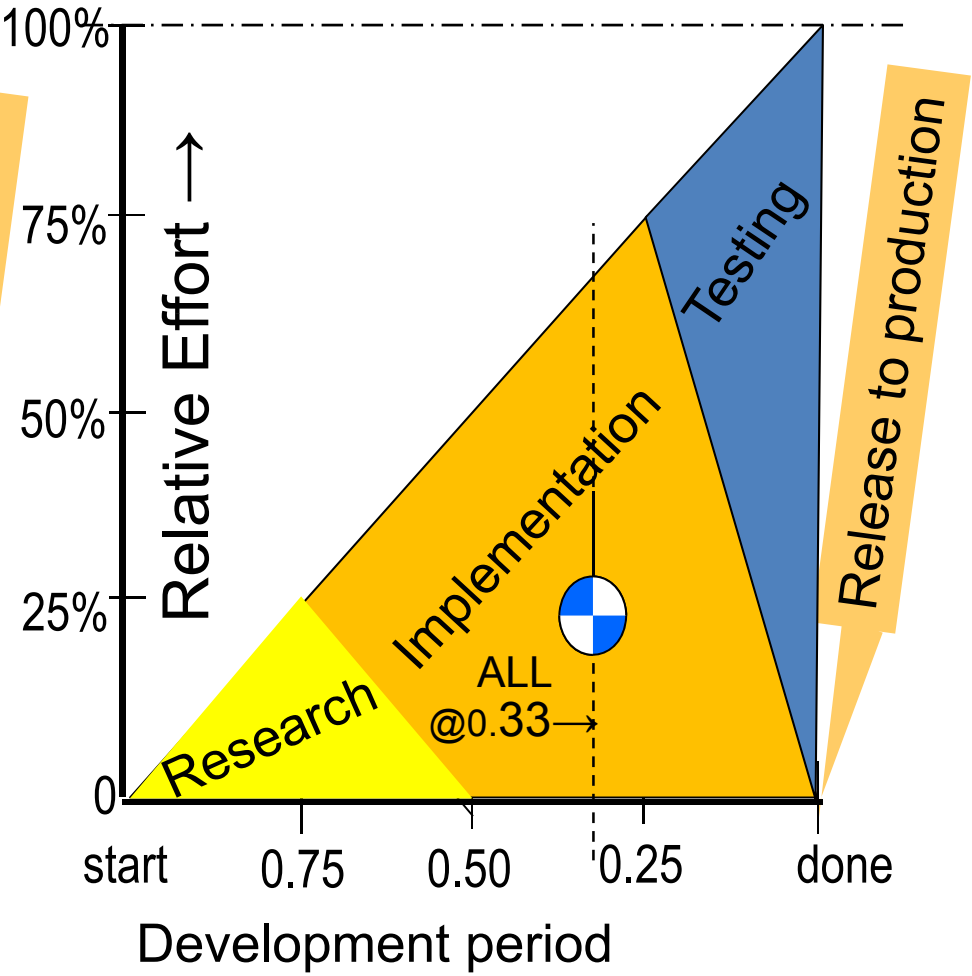
SW Lags


Development done
β
General availability
Sales
RTP
GA
development lag
includes testing
&Test ing
Research, Implementation
Centroid of revenue
sales lag
Revenues
Costs
Costs
Centroid of pre-sales marketing costs
marketing lag
Marketing
time

Simple models

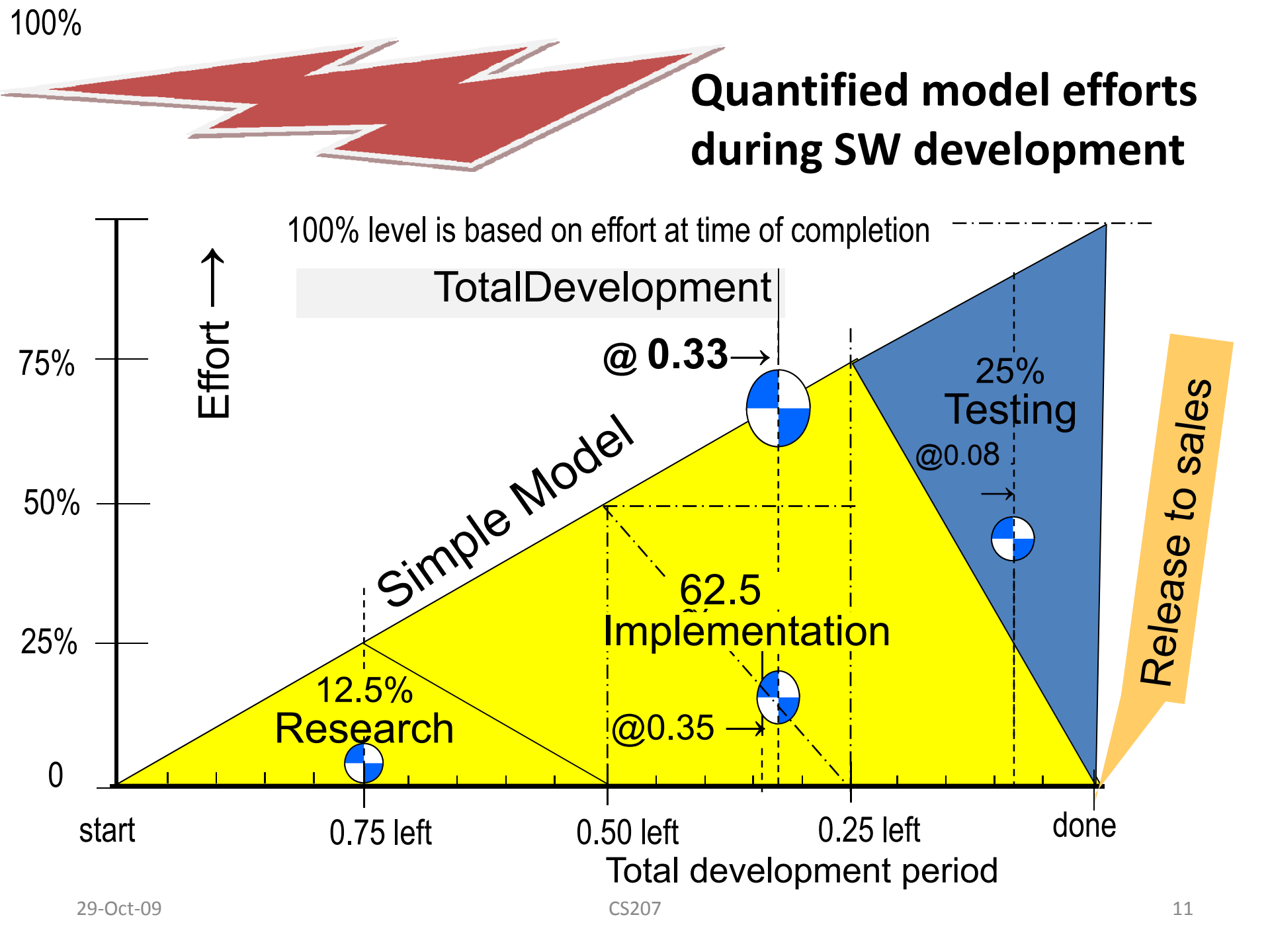


Simplistic model



Simple growth Model

Quantified model efforts during SW development



Start-up development

A startup is unlikely to ramp up linearly

Use exponentional growth, exp 0.025

Assume

1. 12.5% research
   Given that idea is clear, only towards for implementation
2. 25.0% testing
   Minimal and risky
3. 67.5% left for implementation

- Overlap research and implementation until testing starts
- Overlap implemtation and testing until RPS

Results Overall centroid @0.27 before RPS -- later

Research from 1.00 to 0.33, centroid @ 0.65 before RPS

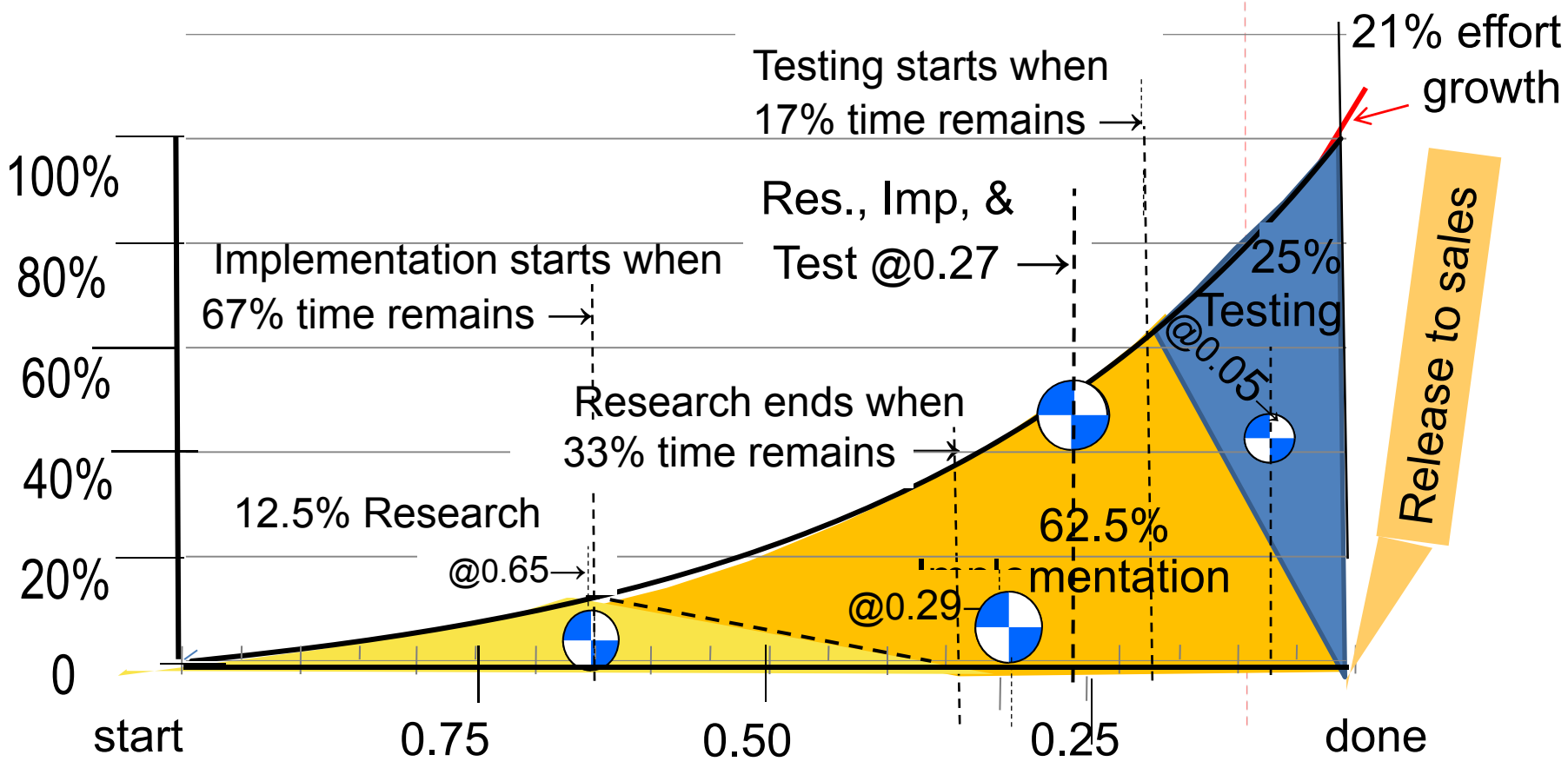
Implementation from 0.67 to 0.00, centroid @ 0.29 before RPS

Testing from 0.17 to 0.00, centroid @ 0.08 before

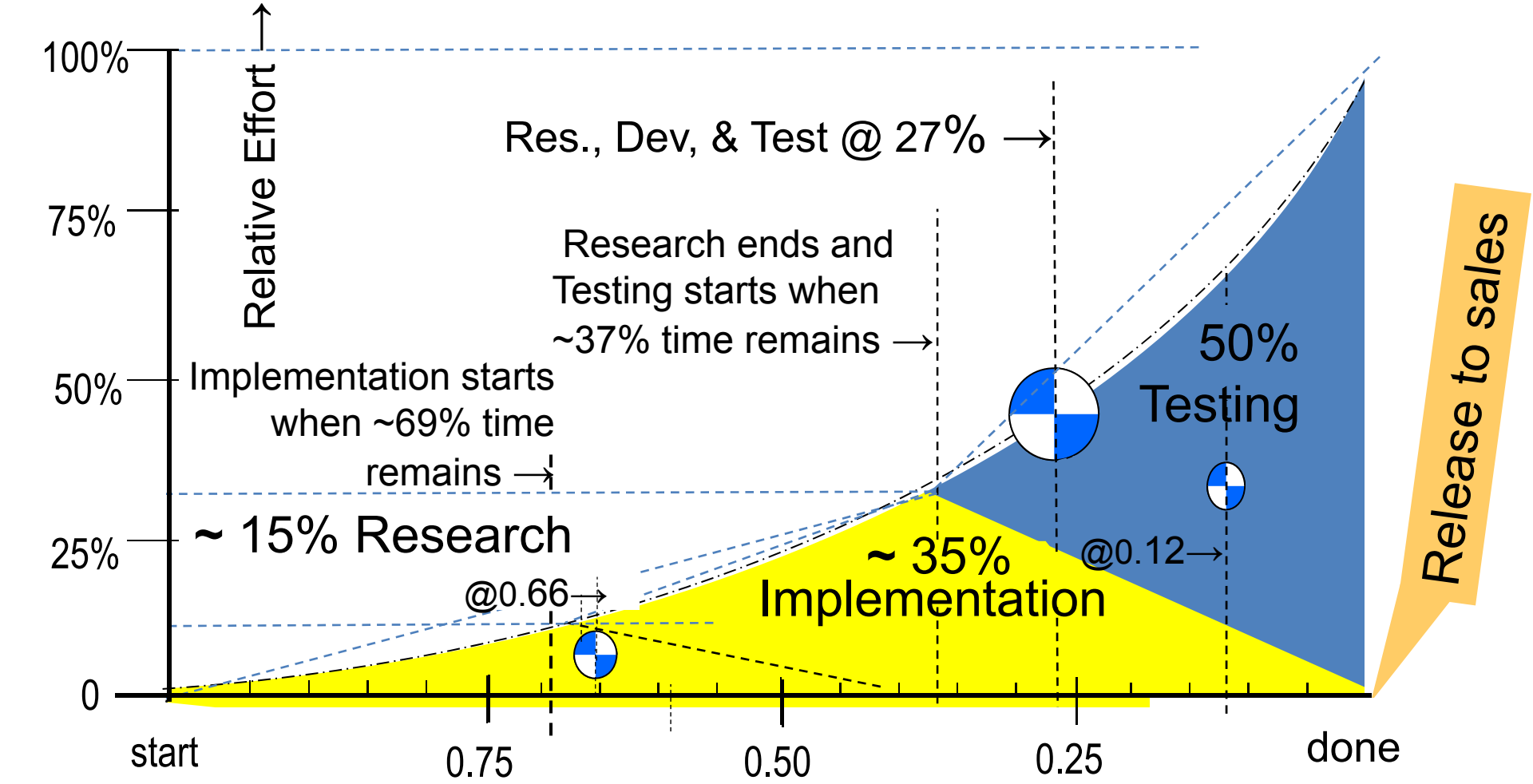
Hiring rate at RPS 21%, at the limit for effectiveness

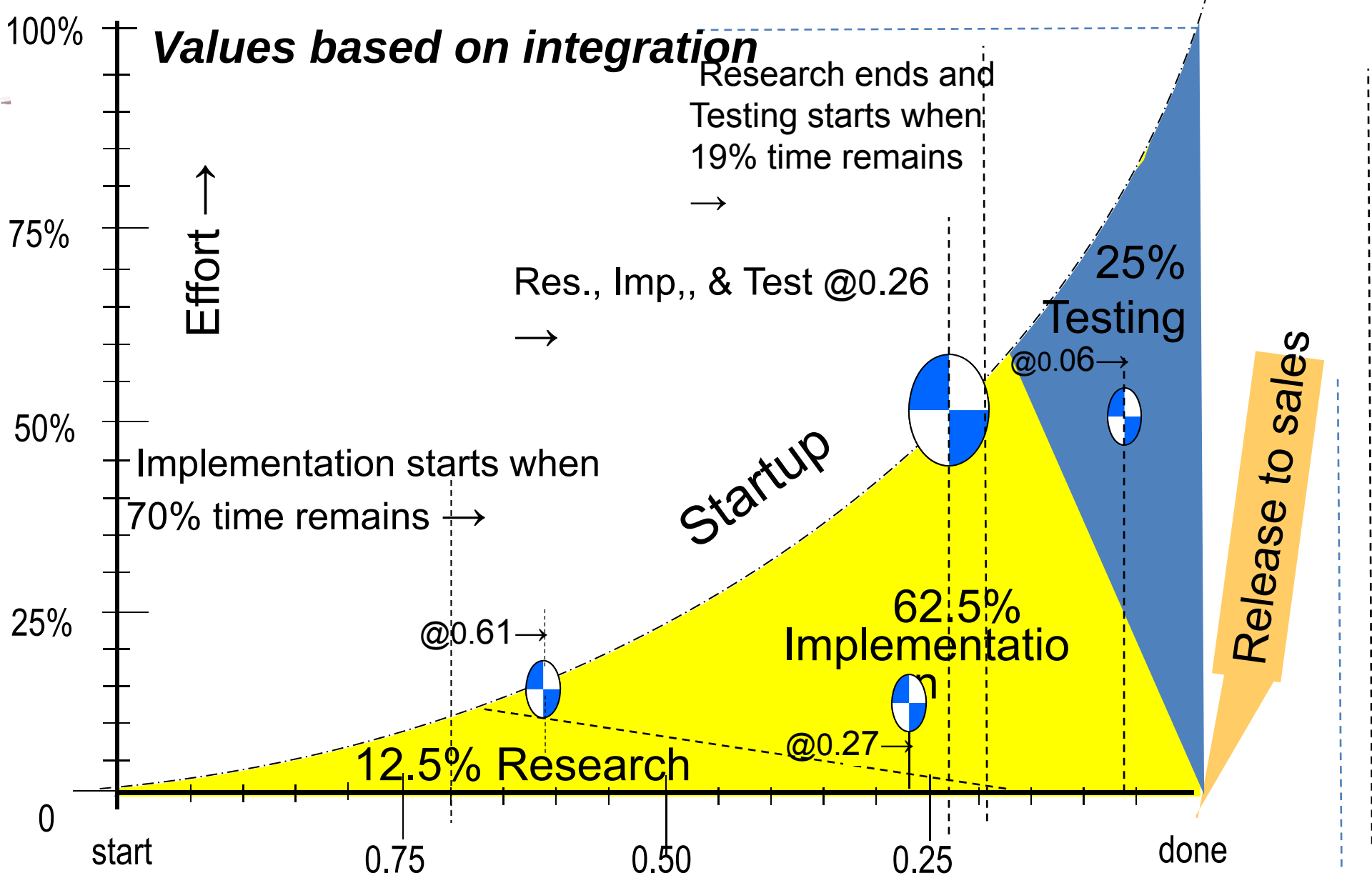
Ignore different staff salaries

Graph of start-up development



Start up with 15% research and 50% testing effort





Start up development with 12.5% research and 25% testing effort

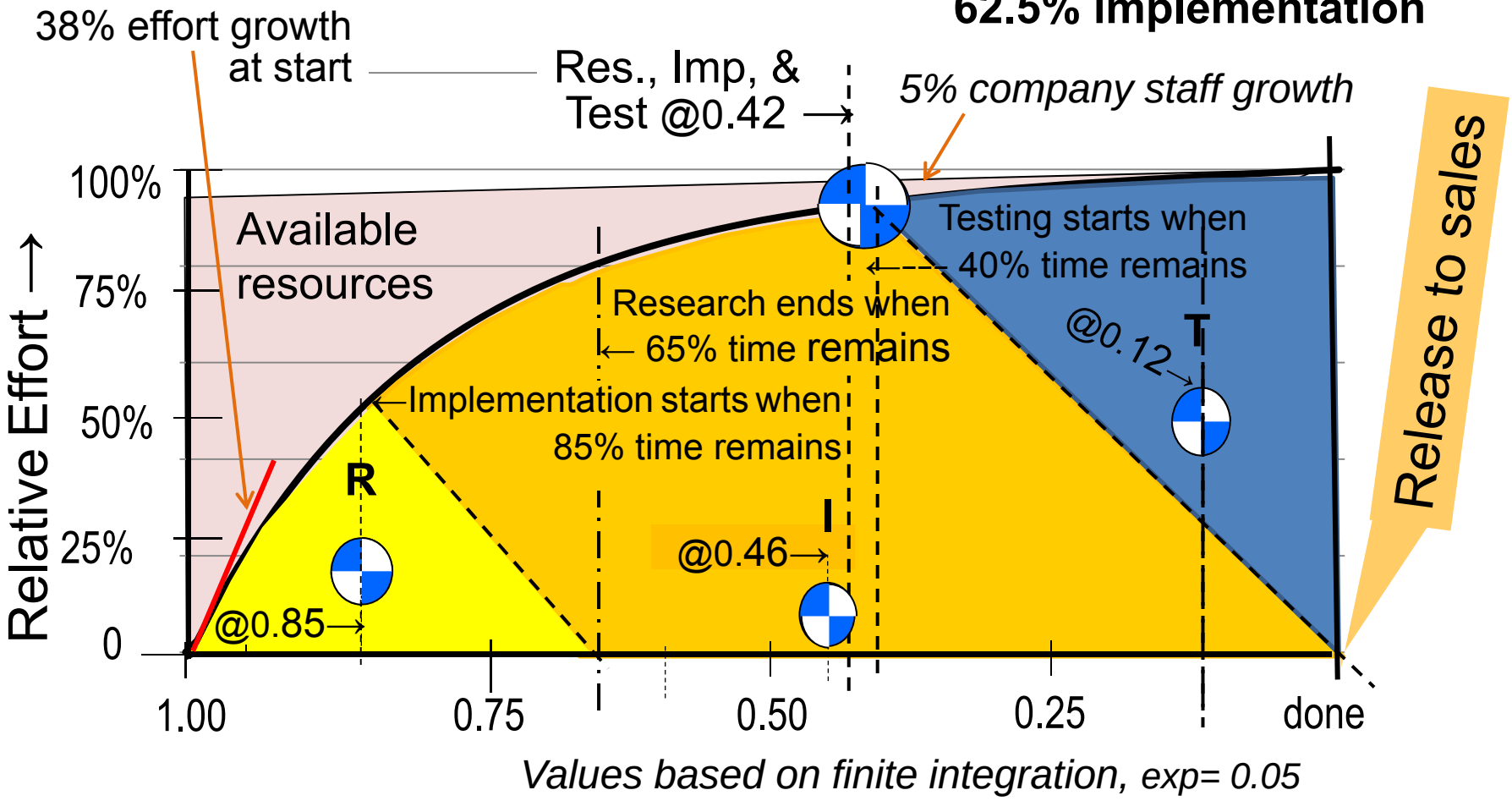
Fig.7i

Development in mature company with

12.5% research and

25% testing effort,

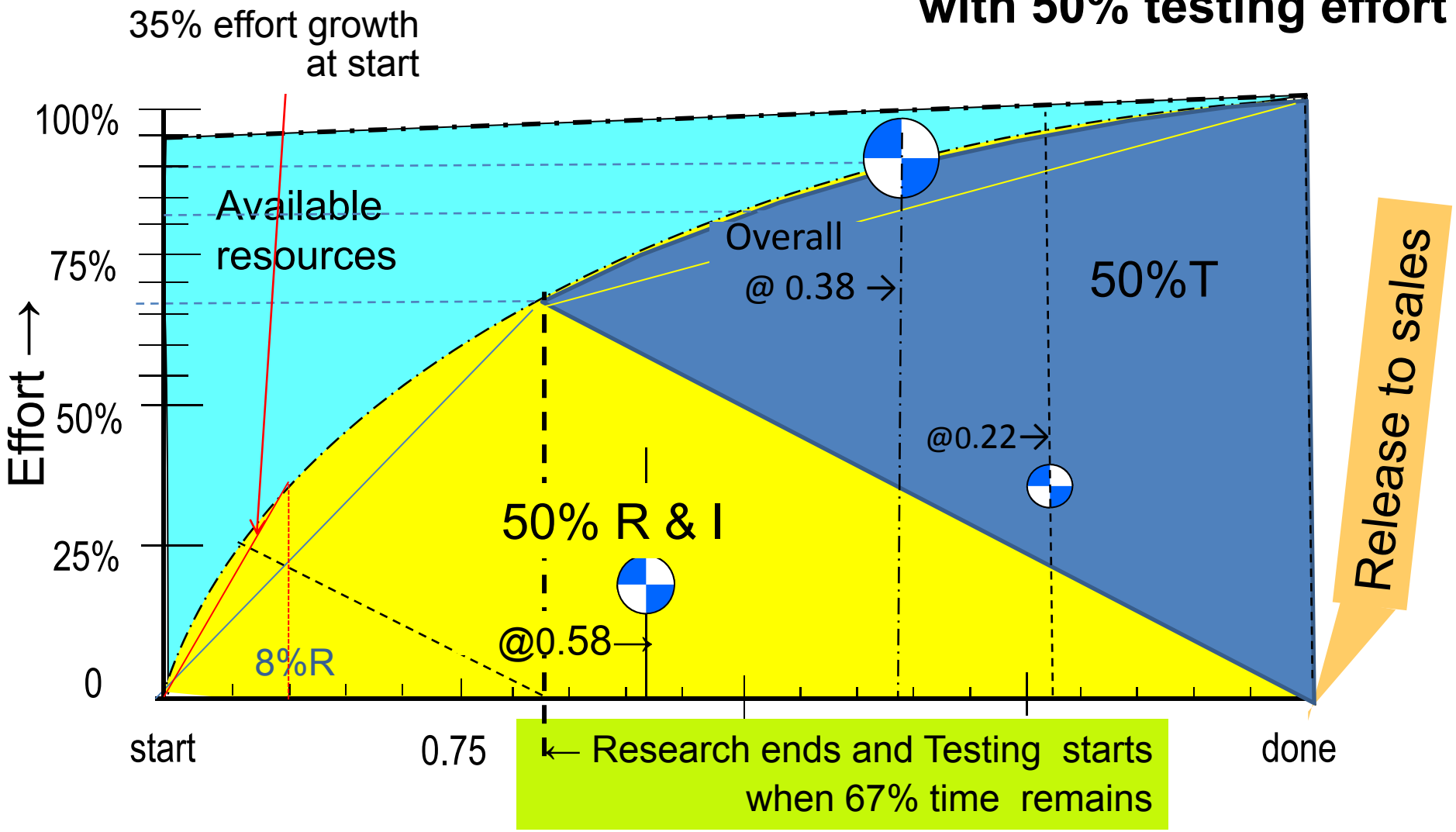
62.5% implementation



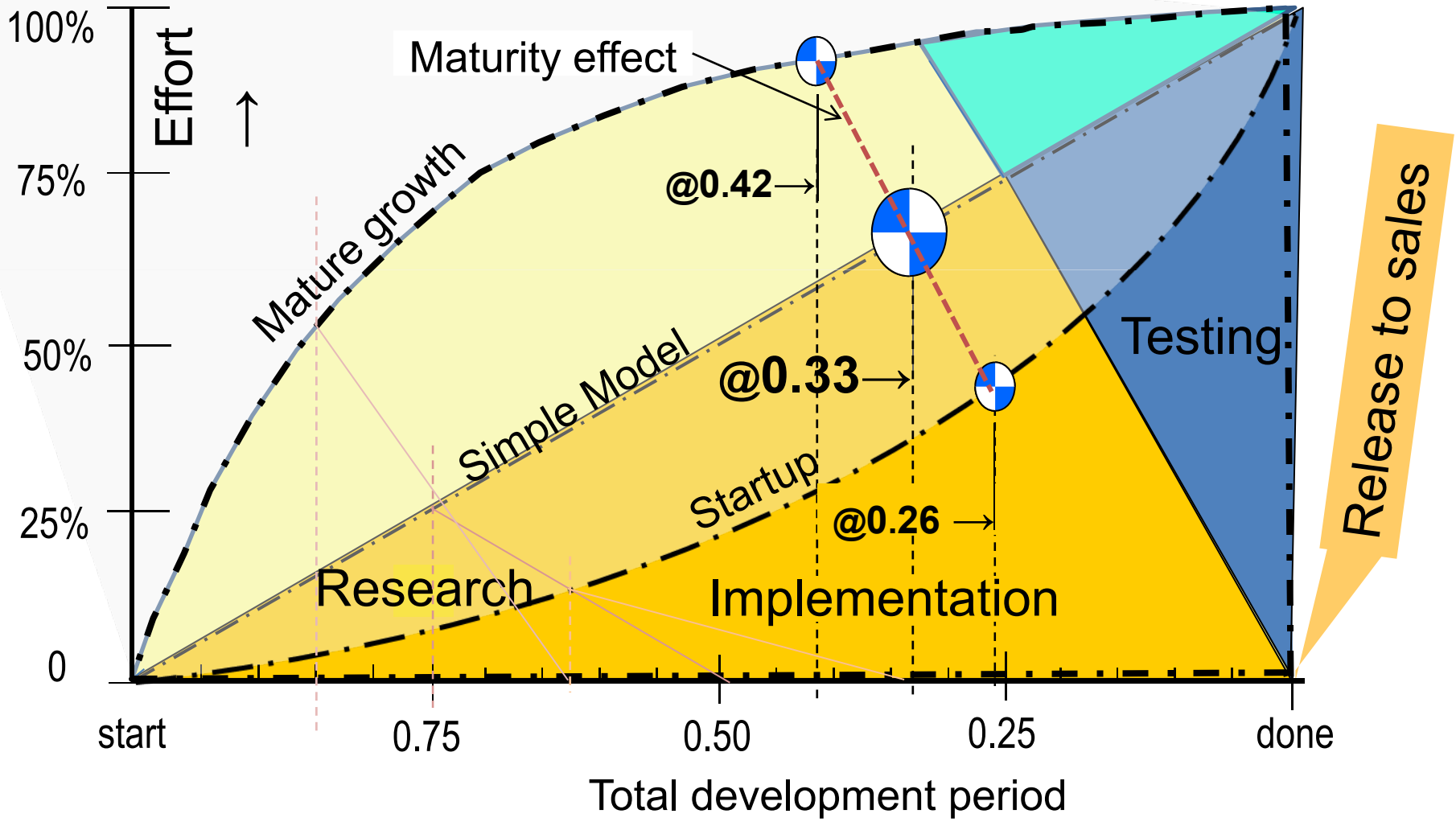
Product revision,

Mature development

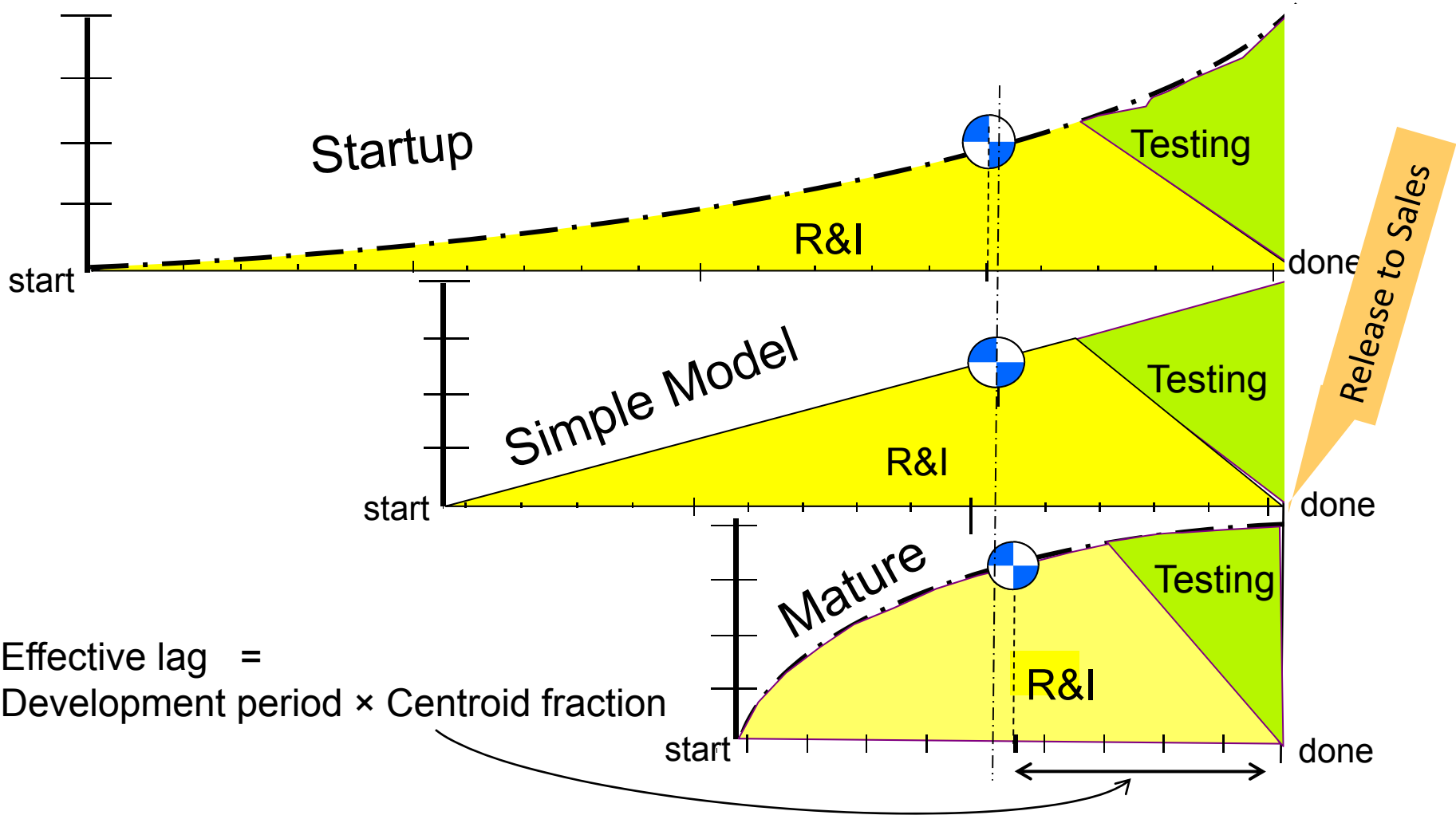
with 50% testing effort



Summary: Maturity effect



Lag differs less than development period



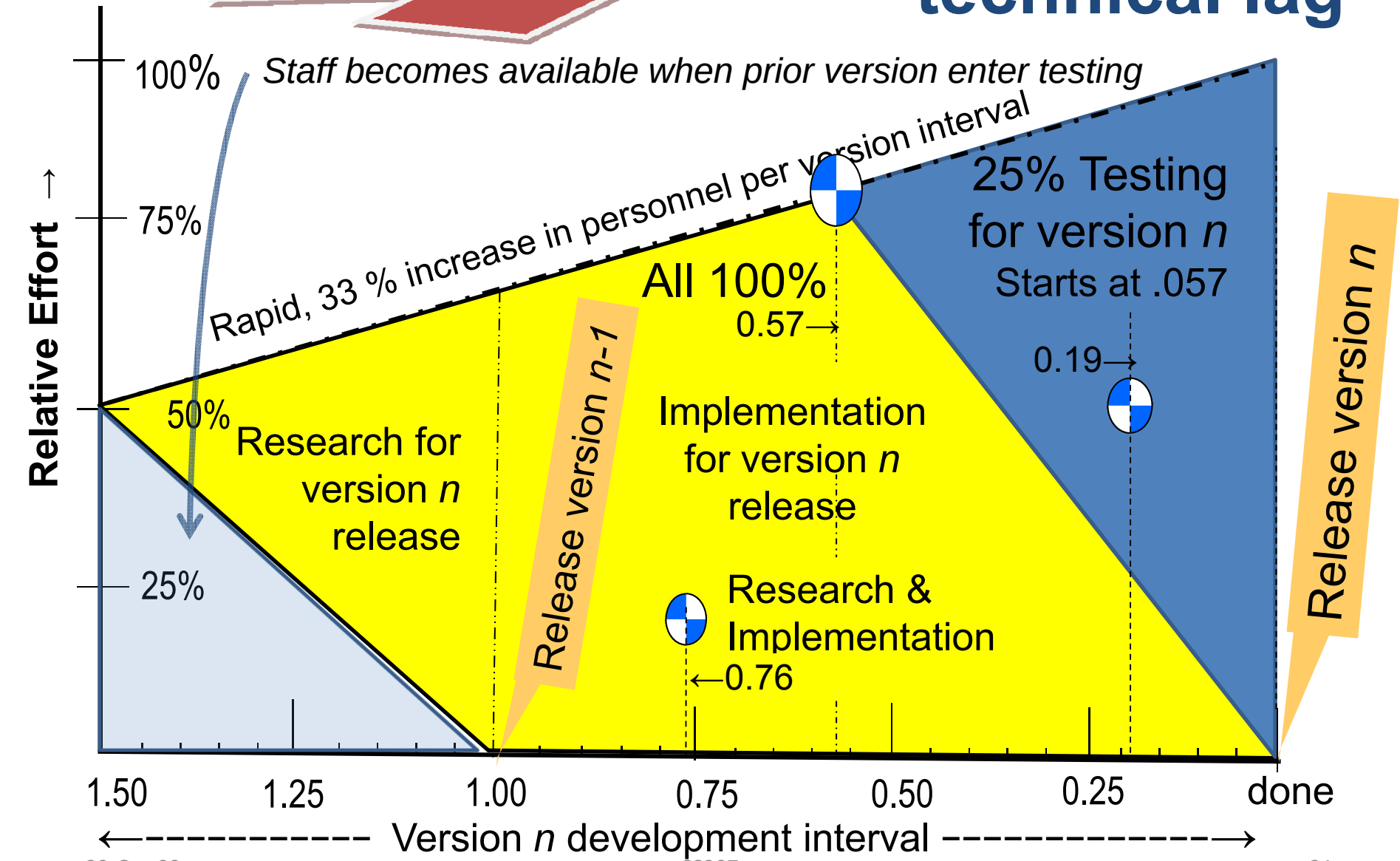
Effective lag =
Development period × Centroid fraction

Ongoing development

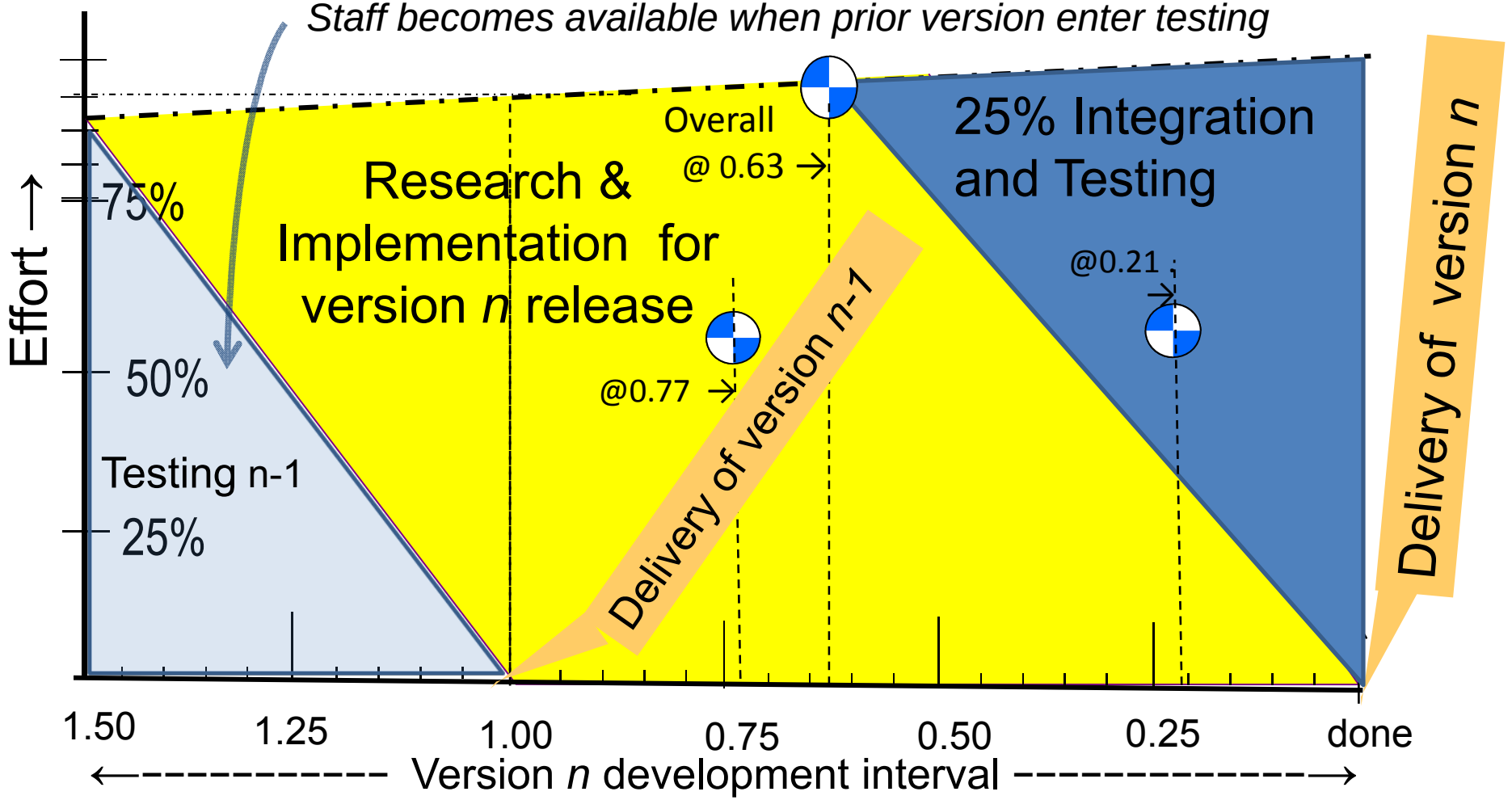
New considerations

1. Have staff already
   a. Early versions rapid growth, but observe ~20% limit
   b. Later, best grow slower
2. Can overlap version development
   a. Don't let valuable staff be idle
   b. Missing features should already be understood
   c. Rapid analysis of problems to allow next version fixes
   d. Any research should be done before major staff effort
3. Adequate testing to keep reputation

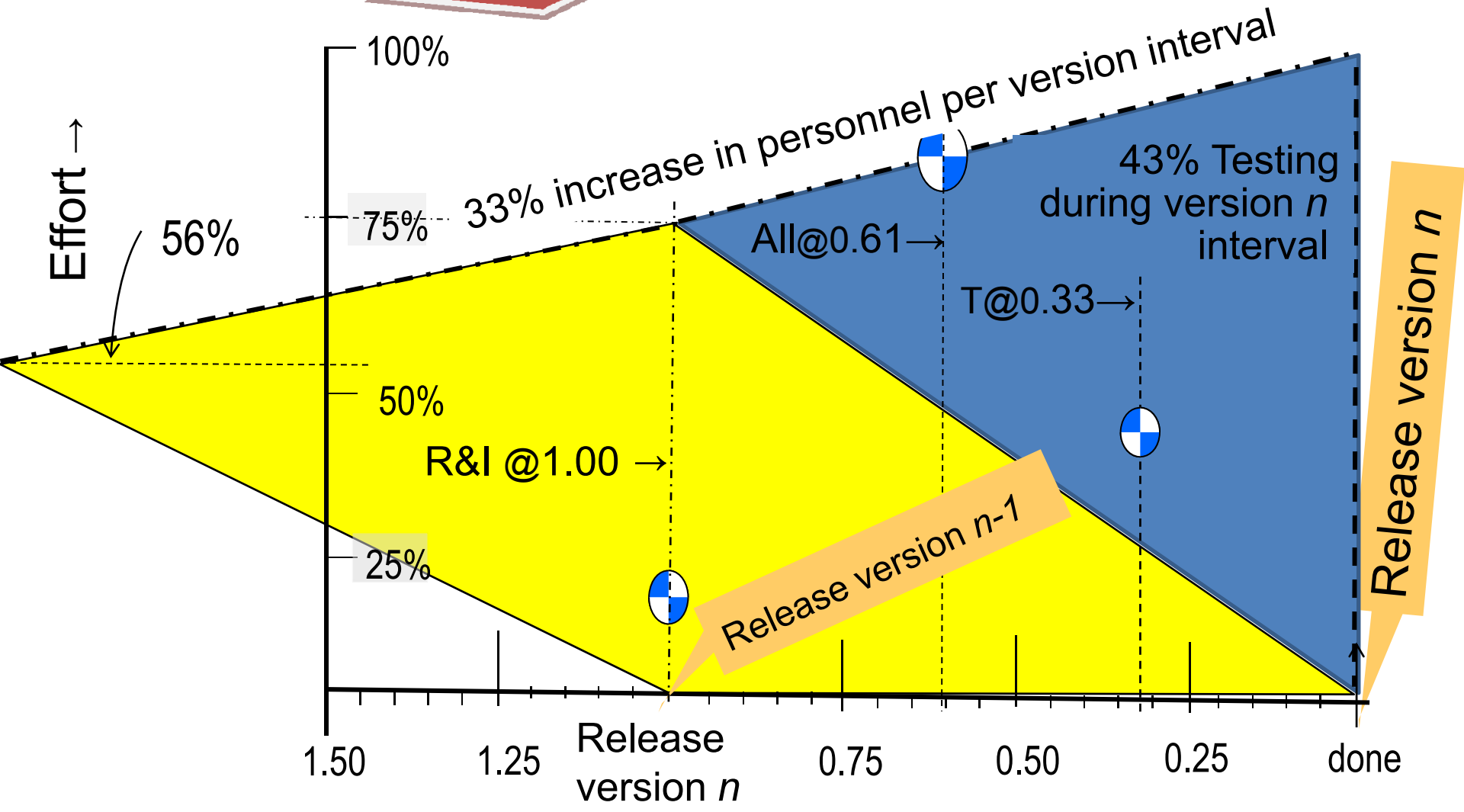
2nd version technical lag



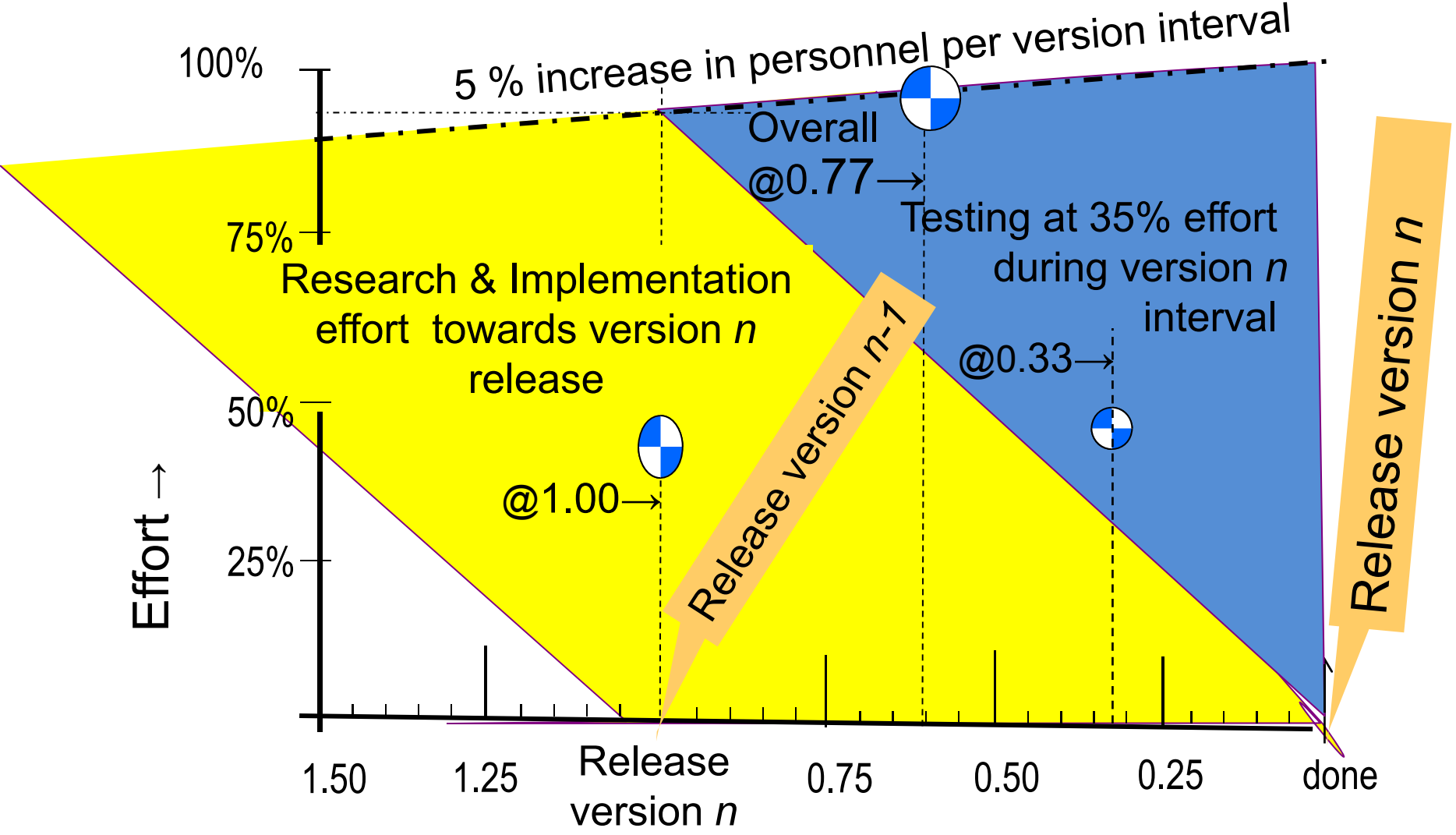
Mature ongoing technical lag



2nd Version substantial testing

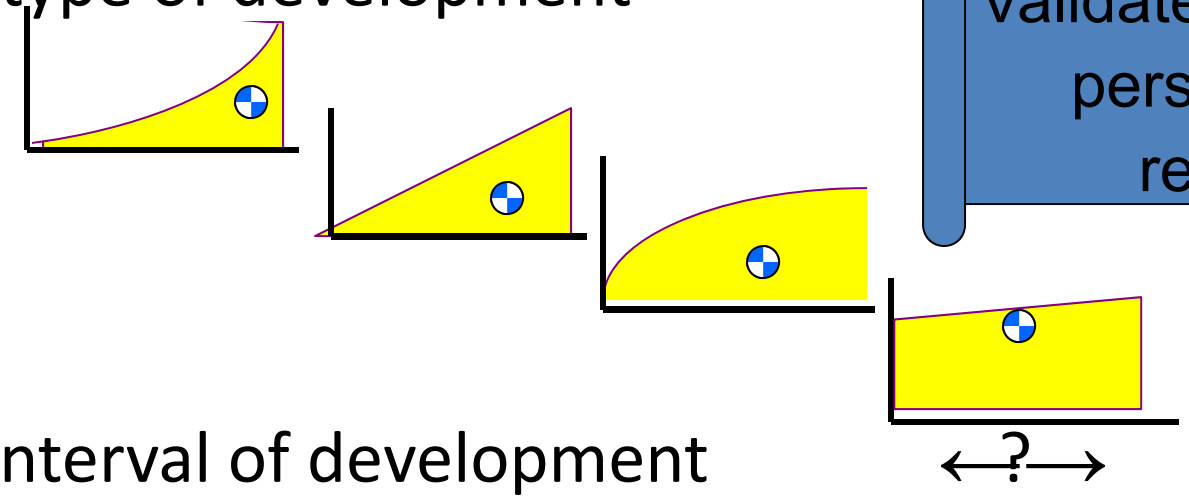


Version development,
mature growth, much testing



Determine technical lag

1. Determine type of development
   a. Startup
   b. Simple
   c. Mature
   d. Ongoing ?



Validate from personnel records

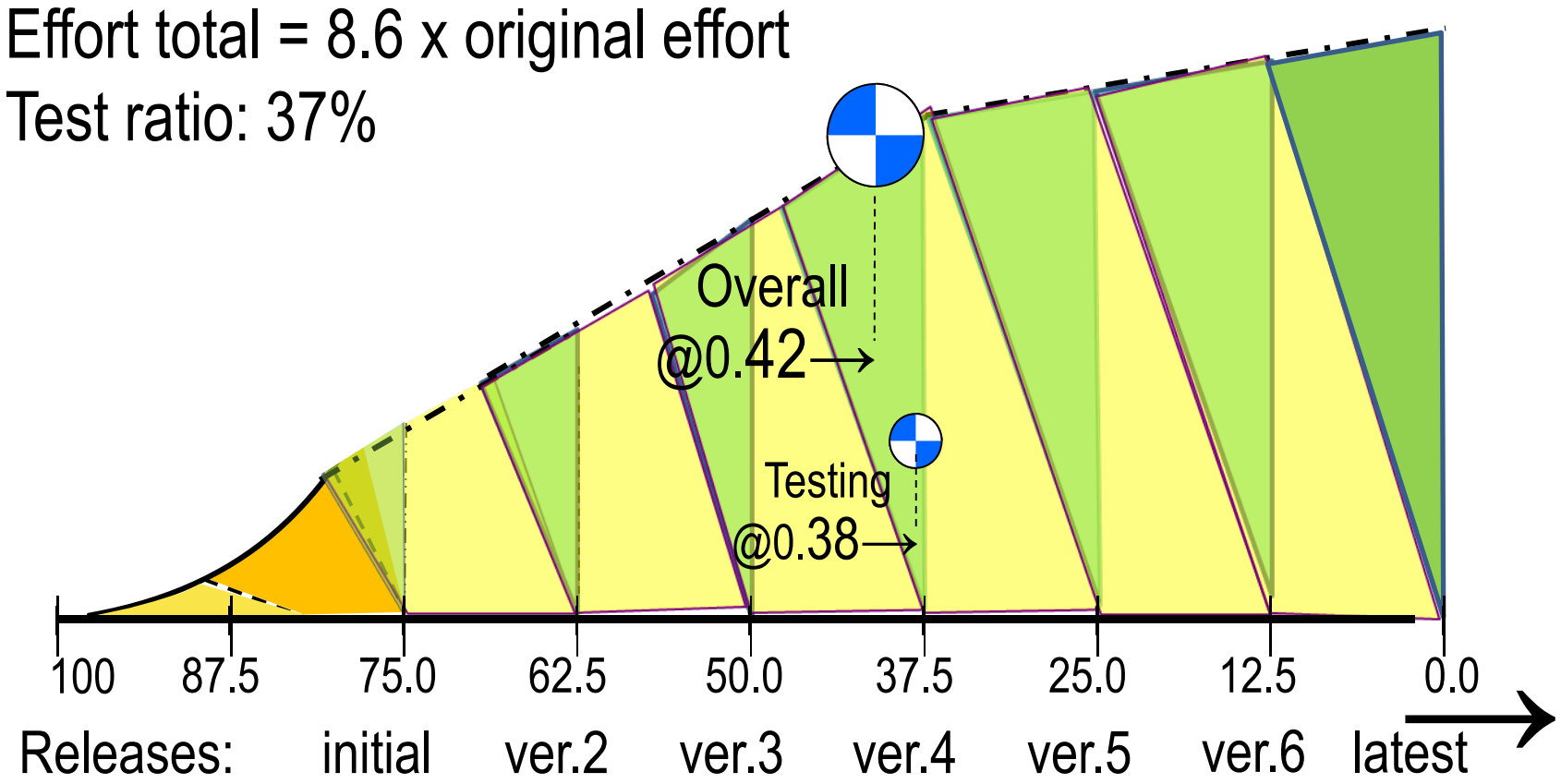
2. Determine interval of development

3. Does testing contribute to IP?

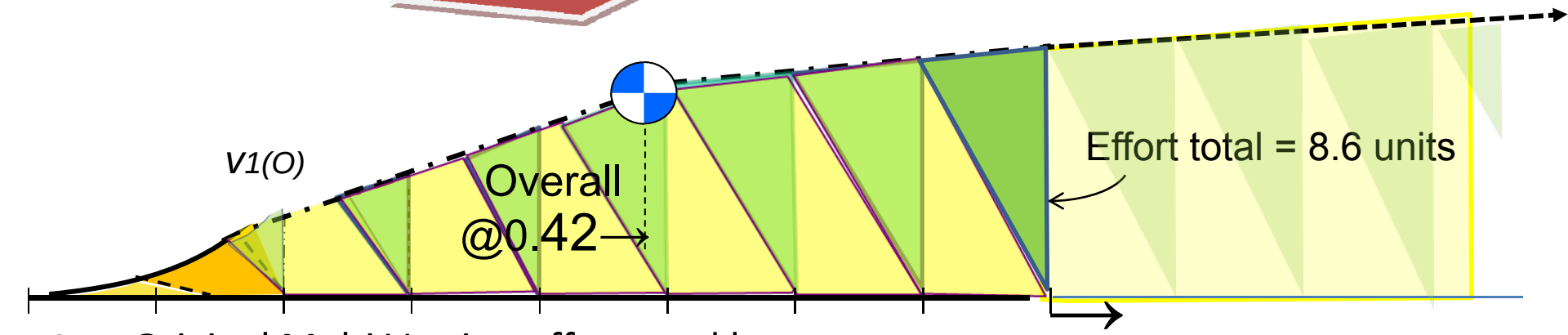
4. Determine growth of personnel effort

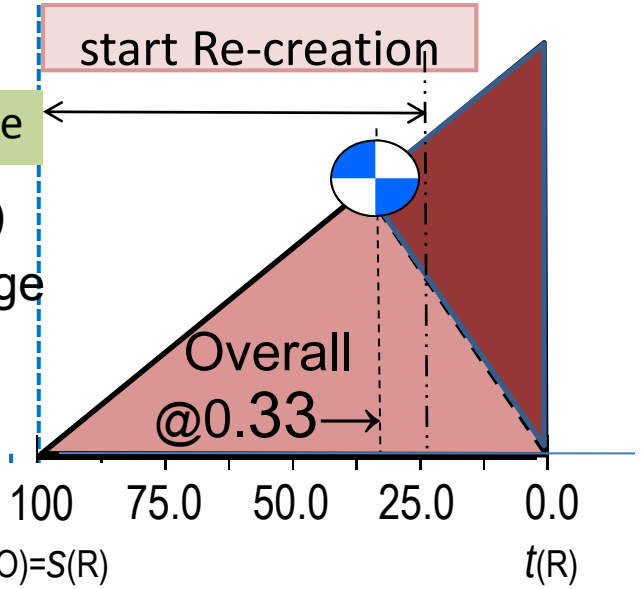
Lag, because it precedes IP diminution, has a large effect on economic valuation

Multi Version product effort and lag



Competition effort and time





Competition *(drawn to scale)*
Growth Rate 20%/year average
Effort total = 5.4 units

Competition/ Original multi version source
Effort ratio R/O = 0.63
Time ratio (t(R)-S(R)/t(O)-S(O)) = 0.41

But at that point the original is 3.5 versions ahead of the competition!

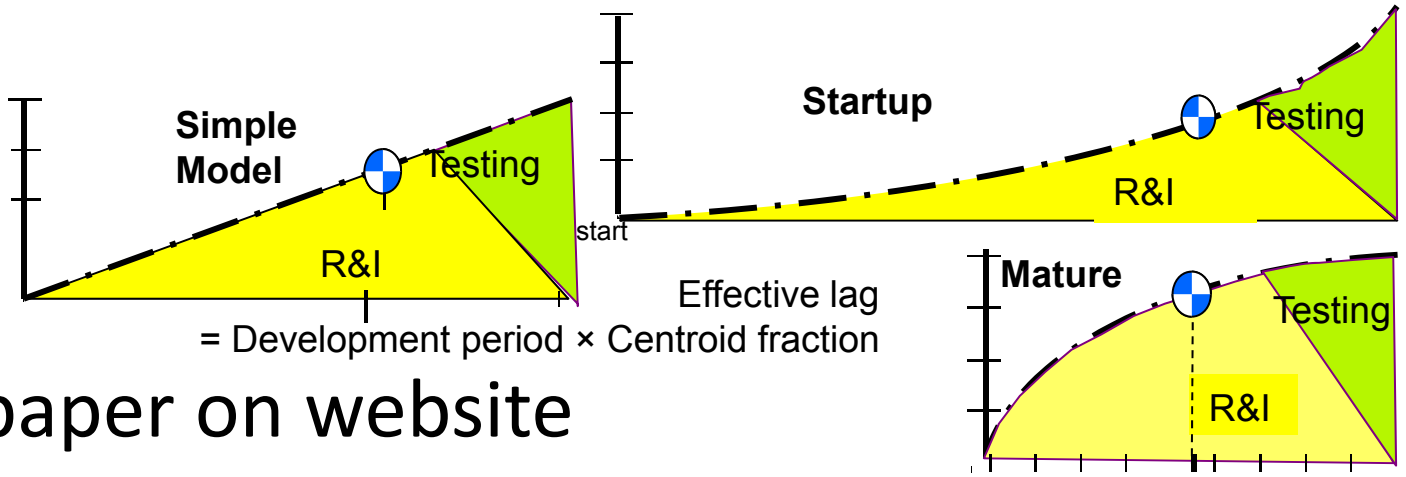
Lag conclusion
(snake in the grass)

Lag is the effective development period

- Has a large effect on early valuations
1. No income during that time
2. A chance for others to overtake you

We assumed a lag of 2 years in the examples.

Depends on product development strategy



Separate paper on website

Discussion

- A long-lived product is hard to displace if
  - It is well maintained,
    - but that becomes costly
  - Keeps up with all standards
- Internal replacement
  - Should be easier
    - But has not been in practice