CS207 #6, 2 Nov. 2012

Gio Wiederhold
Hewlett 103
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.

People & IP

Last week, 26 Oct 2012, Vishal Sikka CTO of SAP

- Focus on personnel
  - Largest cost item in a hightech company
  - Represents the creative part of intellectual capital
- Software is the primary Intellectual property
  - Generated by people
  - Consumed by people internally
  - Embedded in
    - products sold
    - services rendered

Review of layers

Public & Private

Intellectual Resources

Intellectual Capital

Rights owned by the business

Intellectual Assets

Available for transfer

Intellectual Property

Legally protectable

Patents
Copyrights
Trade secrets
Trade marks
Contracts covering intellectual capital

Software users & IP

Companies that

1. develop & sell software → *
   - Basis of IP: income from sales and services
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***

Allocation

1. When there are multiple products
2. When there are other contributors to income
   a. Substantial hardware
   b. Financial consultants in financial firms
   c. Experts in call centers
   d. Brand name

Not all of the income can be allocated to software

- Pareto Optimum

Pareto Optimality

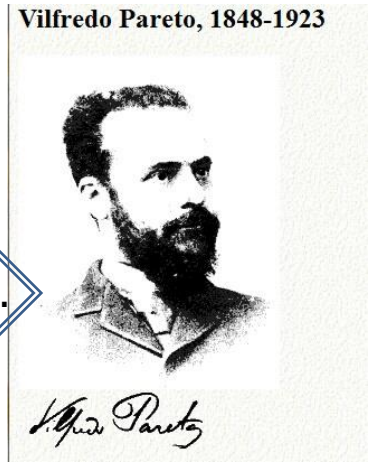
(not Pareto Efficiency : 80/20 rule)

The point were any change lowers the total benefit/cost

1. Spending more on software will have less benefit than spending on other stuff
   a. People
   b. Hardware
   c. Advertising

1873

1870 startup: Rome Railway Co.



2. Spending more on people … lowers the total benefit/cost

Conclusion:

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

Management

- Responsible for optimal operation
- Invest to obtain greatest benefit
- Lags differ
  - People 3 months to a year
  - Computers 1 months
  - Communication 3 months
  - Cloud services
  - Acquisitions 3 months to 3 years, or never
  - Outsourcing contractors 3 to 6 months

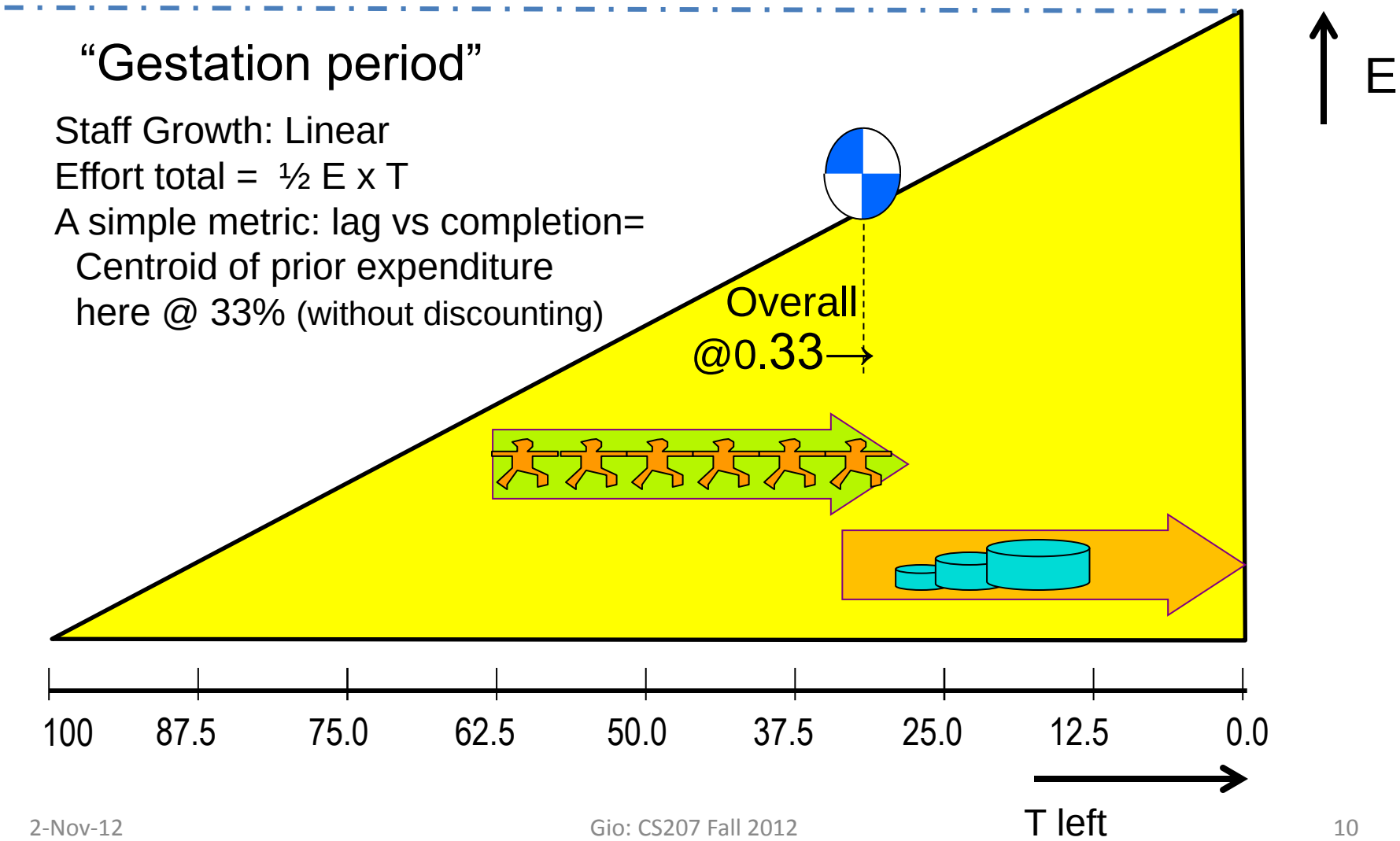
Review Allocation

- For the Pareto-optimality allocation of income one simply uses cost.
  - ➢ But recall: Do NOT use cost as a surrogate for value, value of intangibles come from income.

1) Some spending supports routine income
   - ❑ computers, facilities, cloud services, outsourcing …
2) Other spending supports non-routine income
   - ❖ creative people: engineers, programmers, marketeers
   - ❖ those should be valued, taking leverage and lag into account

Approach: only consider 2) the `Residual' ←economists' term

Lag measures Prior effort



Marketing

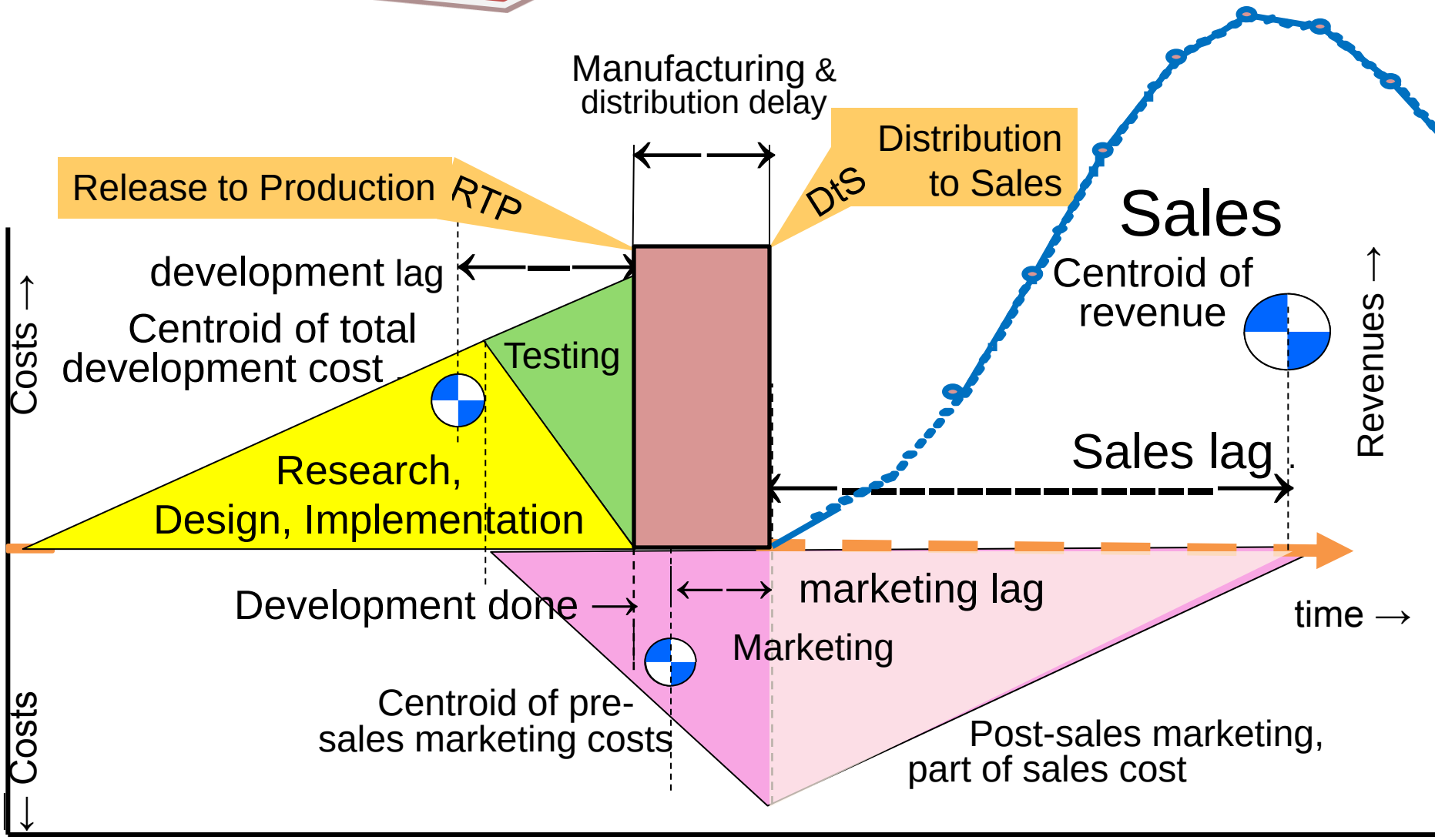
- Business model must allocate spending optimally
  - Technology, as needed, long life and lag

&

  - Marketing, necessary, less lag, slower growth
    - For large 10 IT companies the average value allocated to their brand name is 22% (BW survey).
- Interdependence viral
  - Consistent
  - Relevant
  - Linked by a common name and label
  - Honest ~~FLASH~~

CS207

Various Lags



Expense leverage

A valuation based on cost

1. Collect the expenses e_i over the total lag period p
2. Adjust past costs by a capitalization rate d, $a_i = (1+d)^{p-i}$
3. For year $i = 1 \rightarrow p$ estimate the R&D retained $r_i = 1 - 1/p$
4. Aggregate retained to the end date, $R = \Sigma\, r_i \times e_i \times a_i$
5. From experience, publications obtain an expected leverage m; m can range from *1 to 20* ...
6. Expected value of *IP* $V = m \times R$

$m \approx 2$ in the first model we used

But the estimation of m is verrrrrrrrrrry iffy

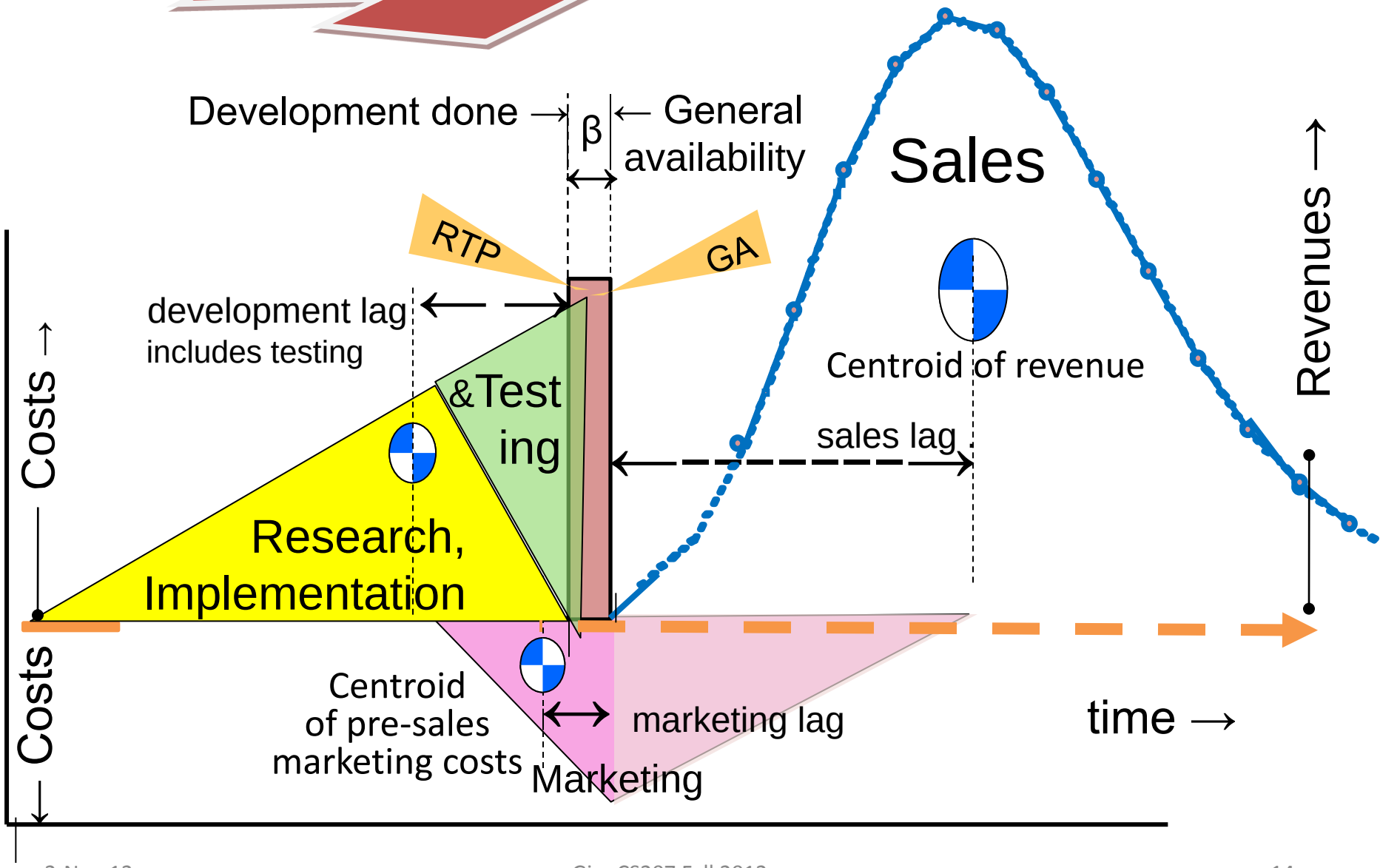
Technological advances are rarely stable

But could be used for a) venture investing m = 6 – 20+

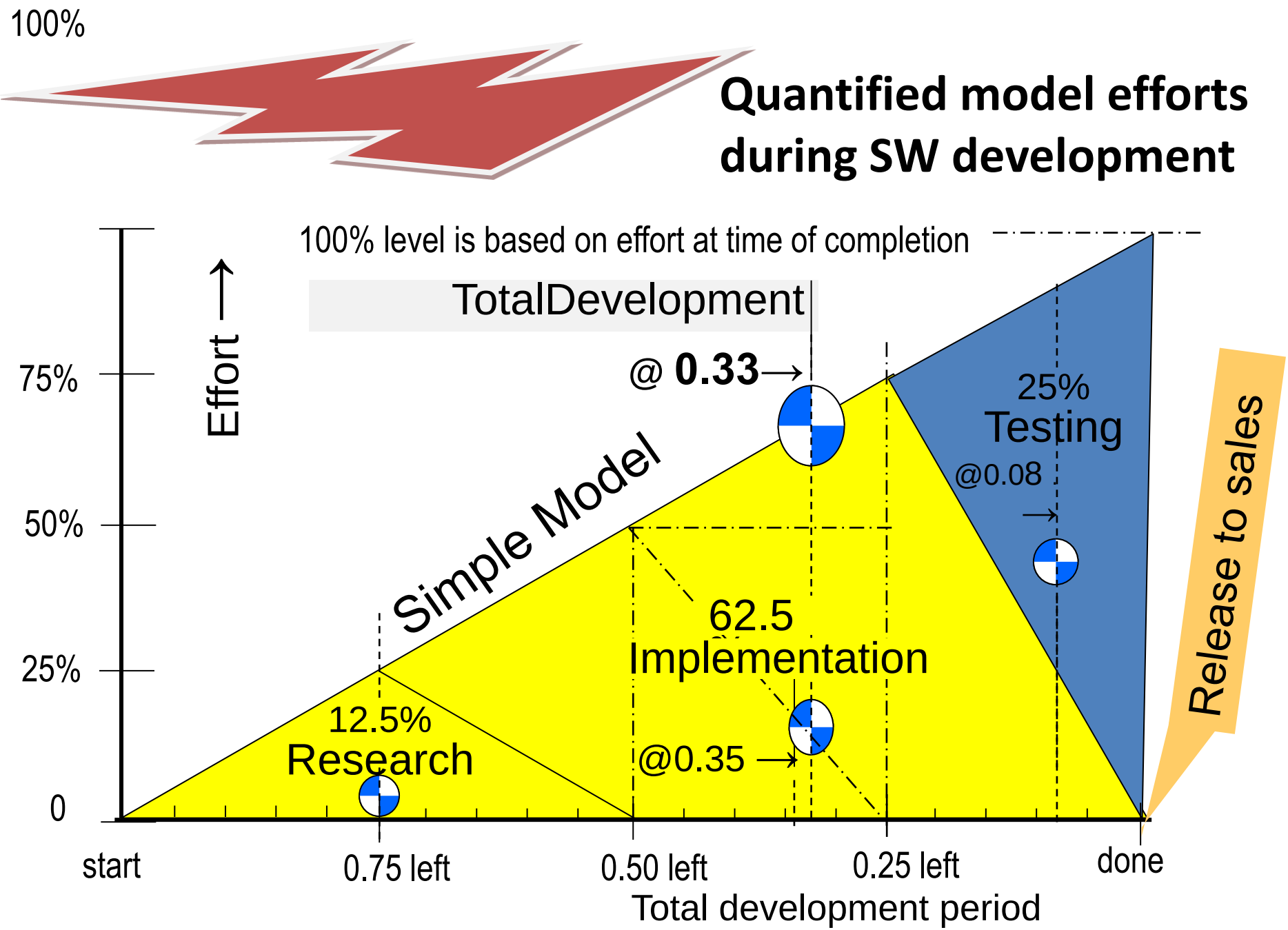
b) advertising -- much untrustworthy data

c) stable maintenance component only

SW Lags


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

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

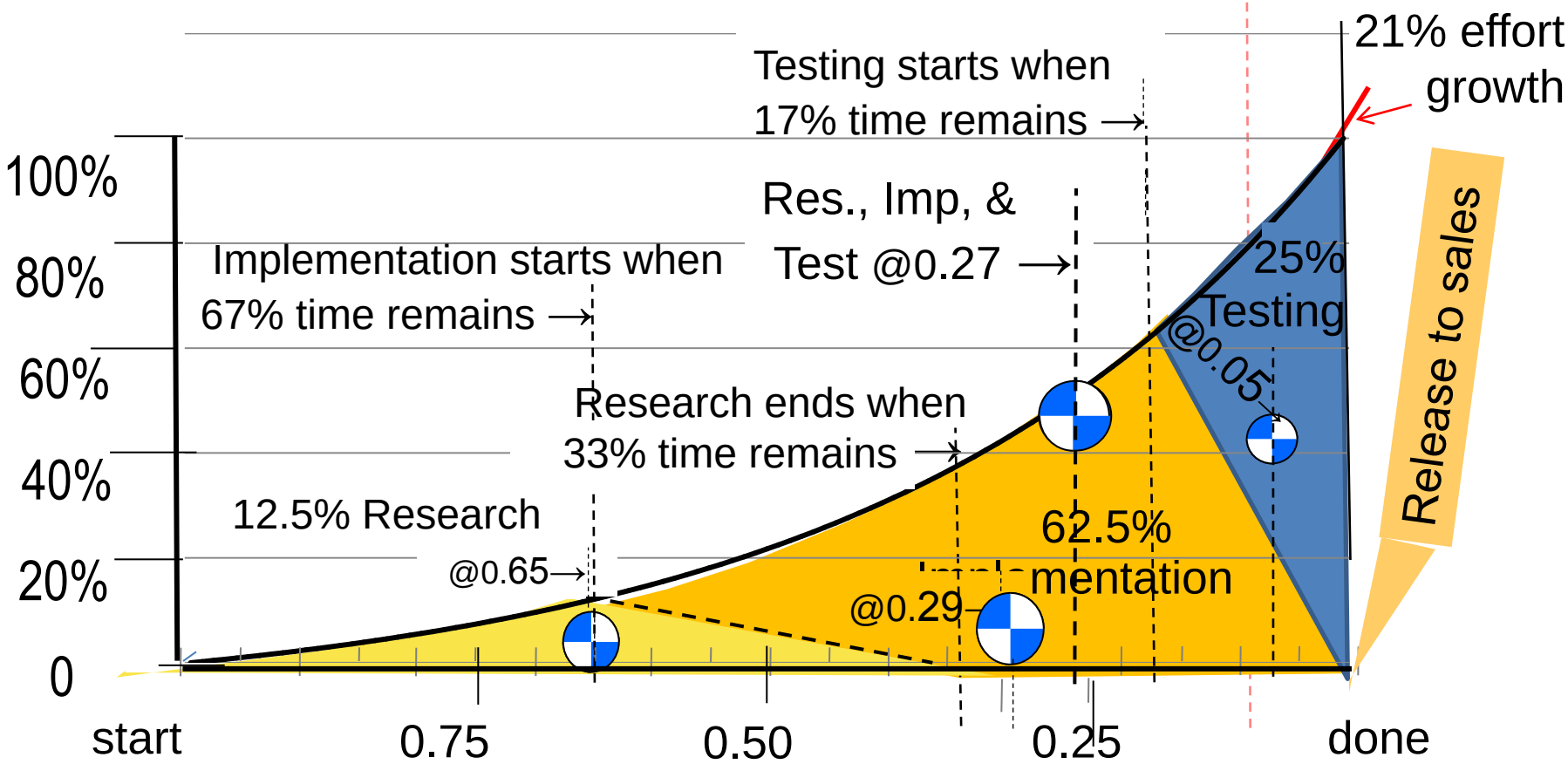
Implementation from 0.67 to 0.00, centroid @ 0.29 before RPS

Testing from 0.17 to 0.00, centroid @ 0.08 before

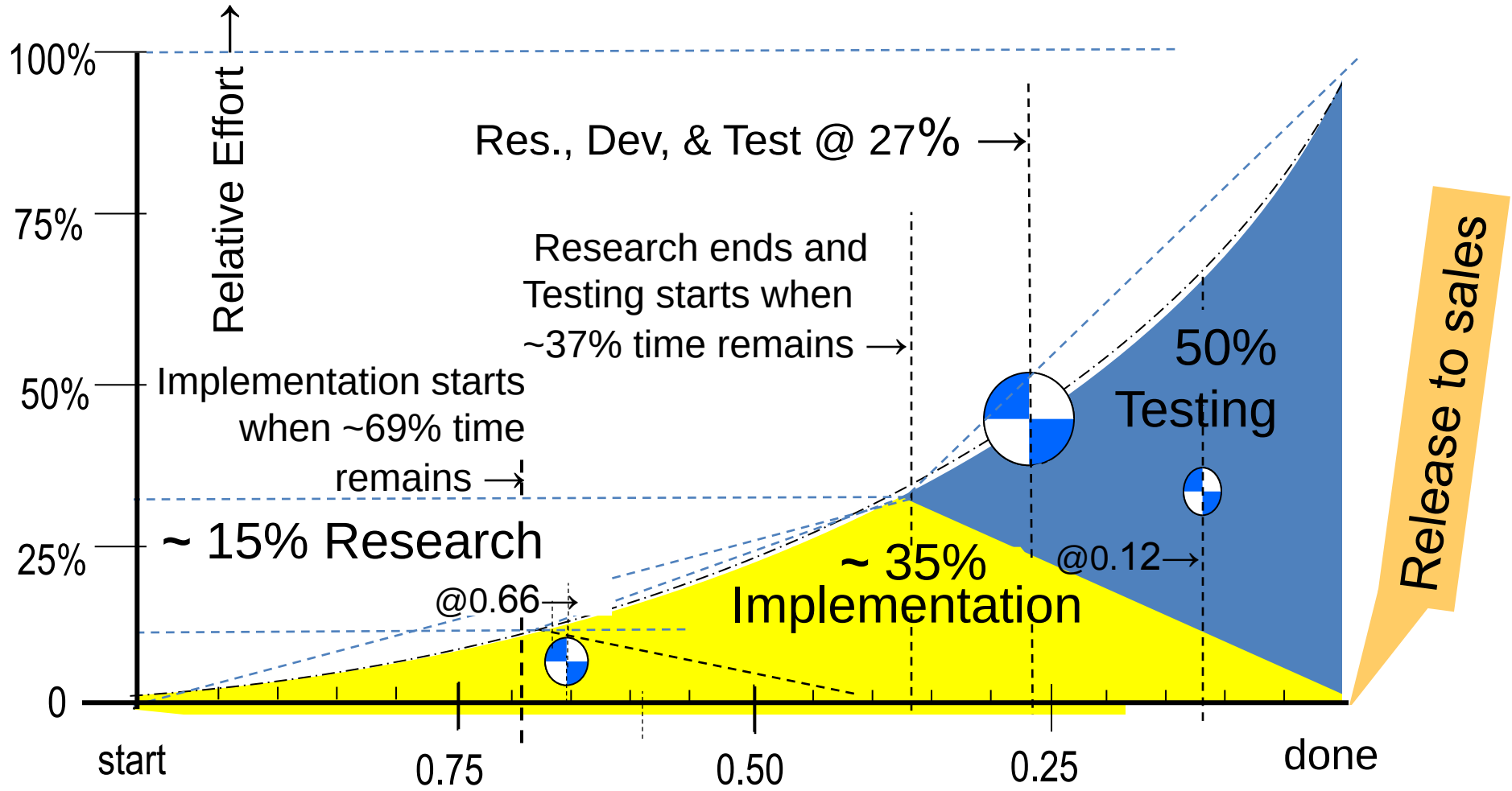
Hiring rate at RPS 21%, at the limit for effectiveness

Ignore different staff salaries

Graph of start-up development



Start up with 15% research and 50% testing effort

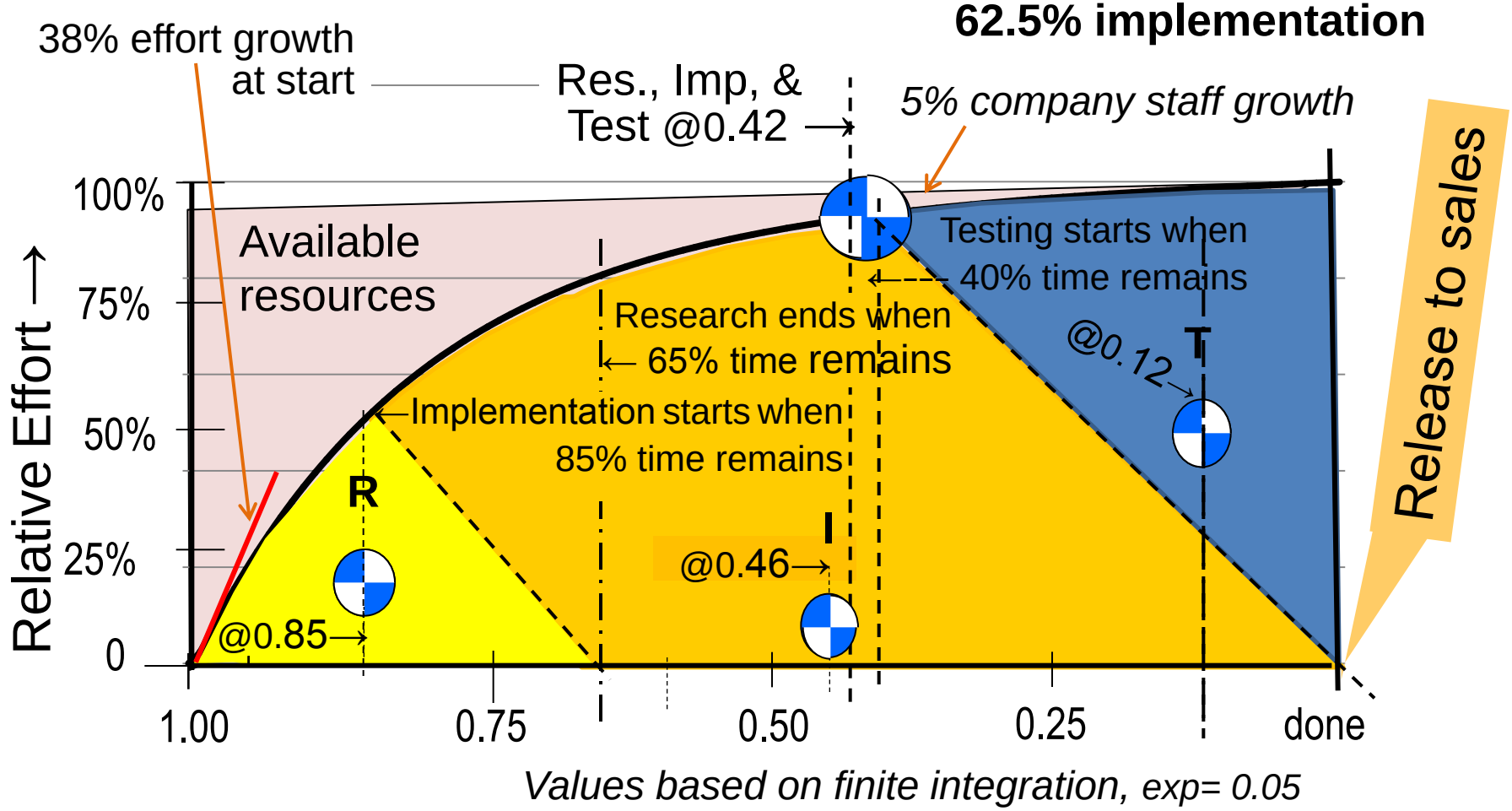


Development in mature company with

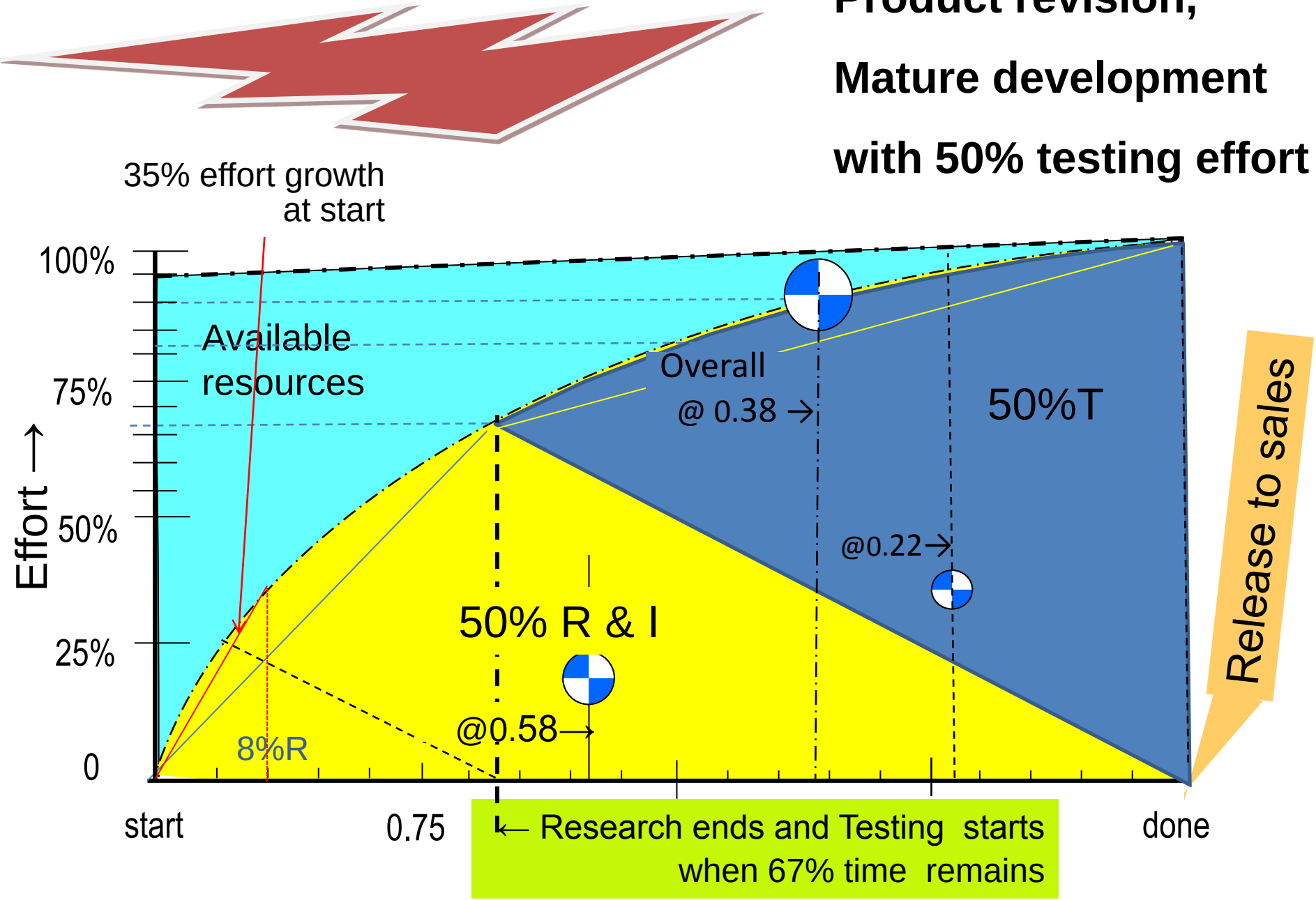
12.5% research and

25% testing effort,

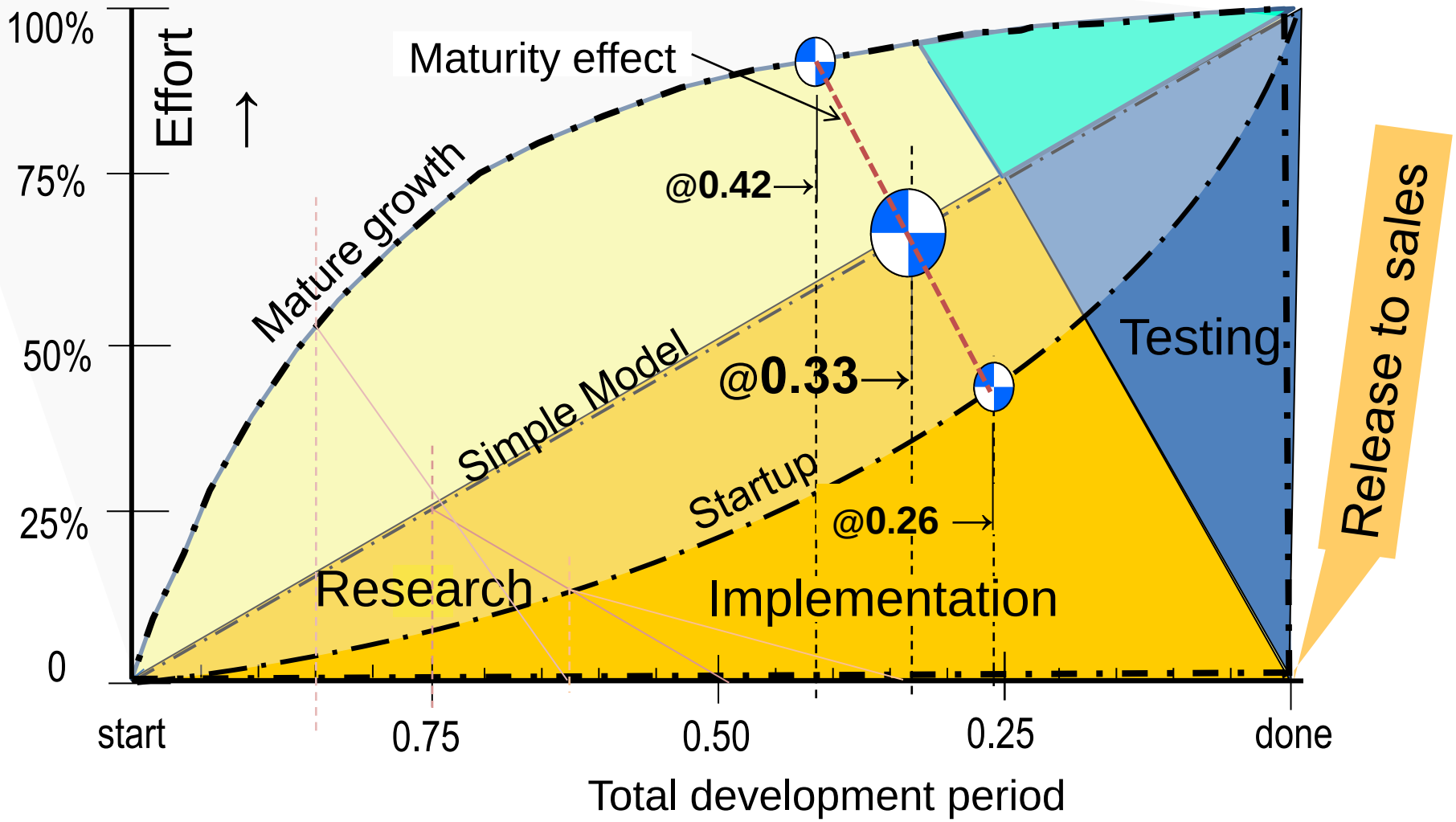
62.5% implementation



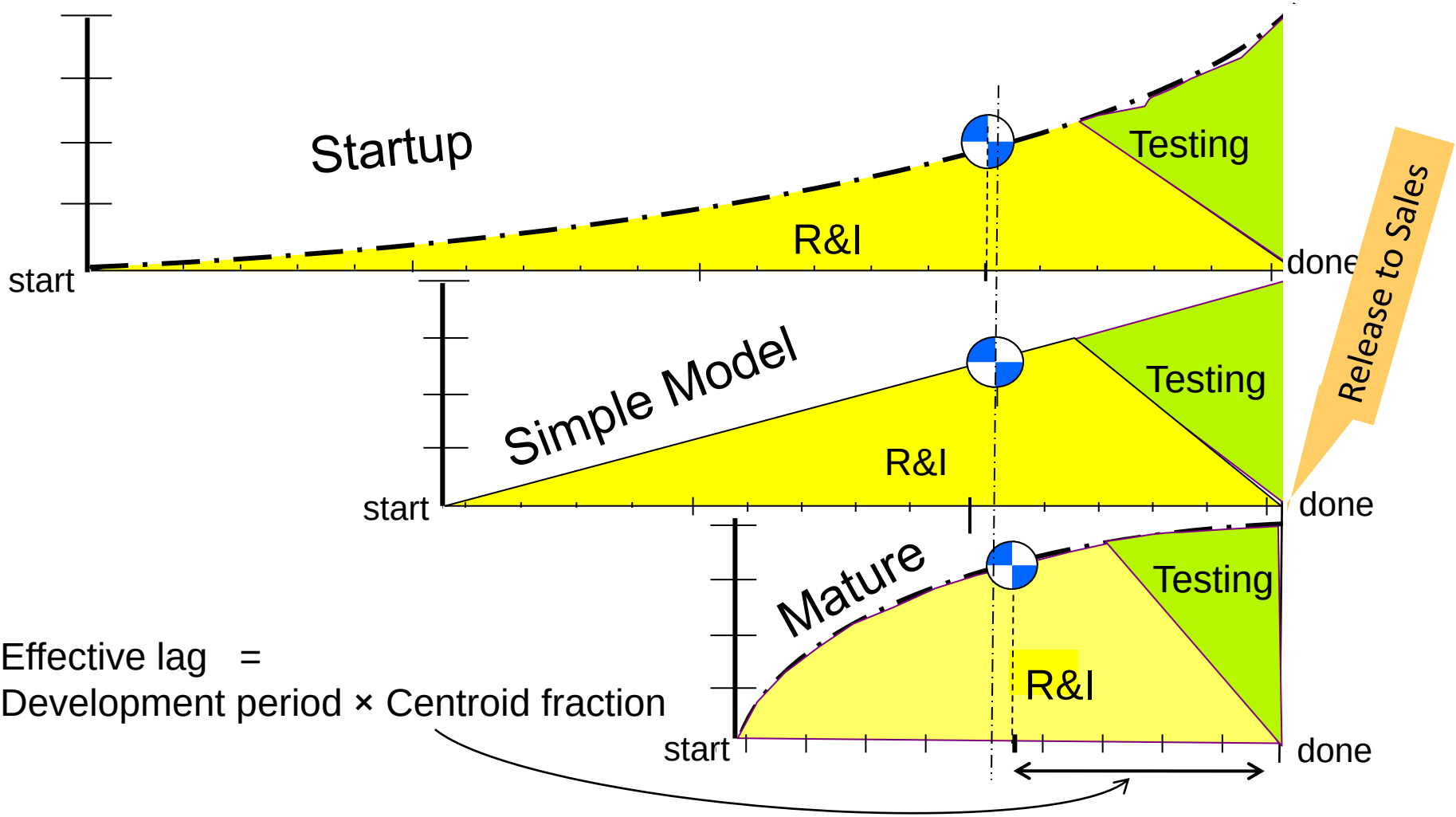
Product revision, Mature development with 50% testing effort



Summary: Maturity effect



Lag differs less than development period



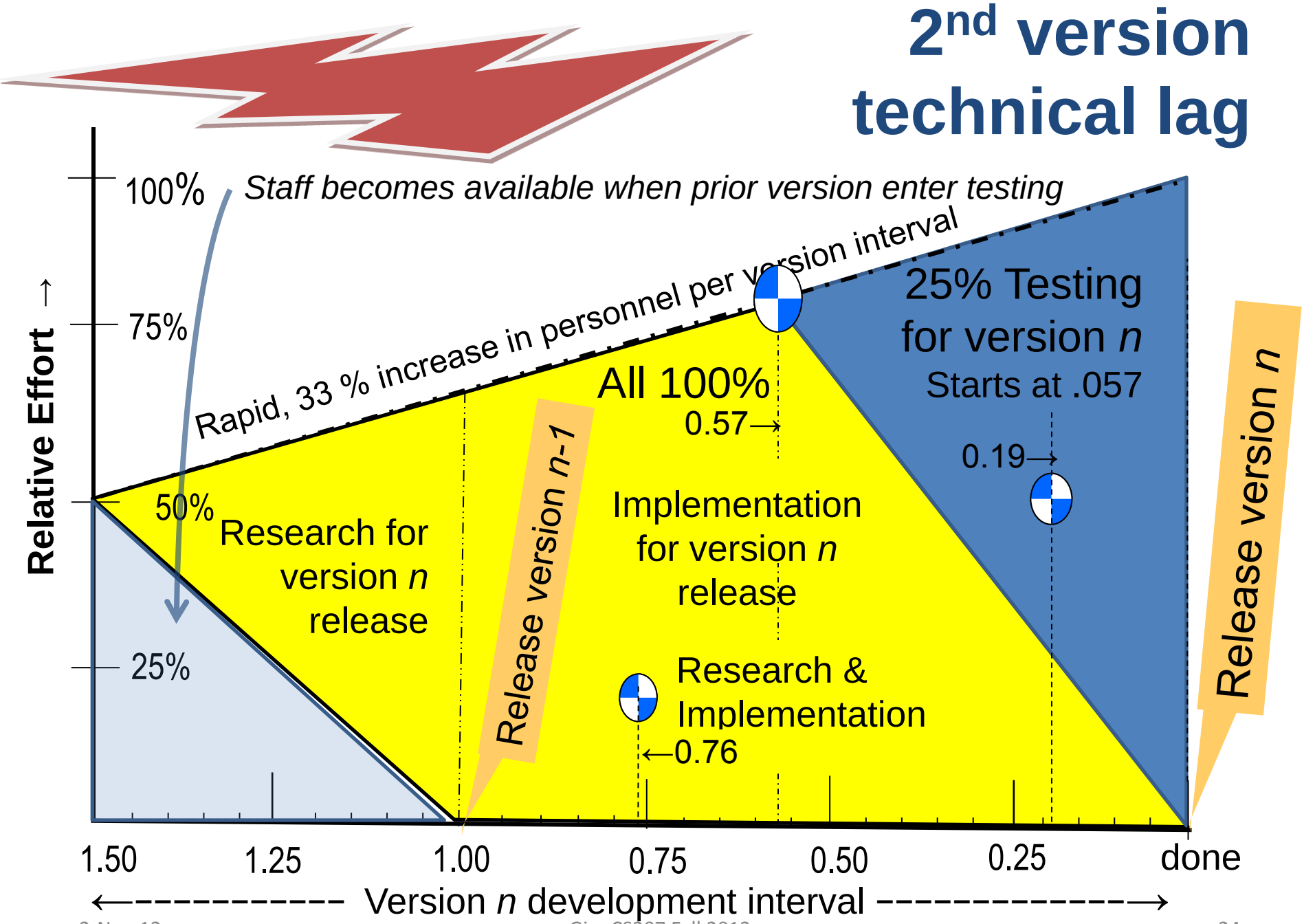
Effective lag =
Development period × Centroid fraction

Ongoing development

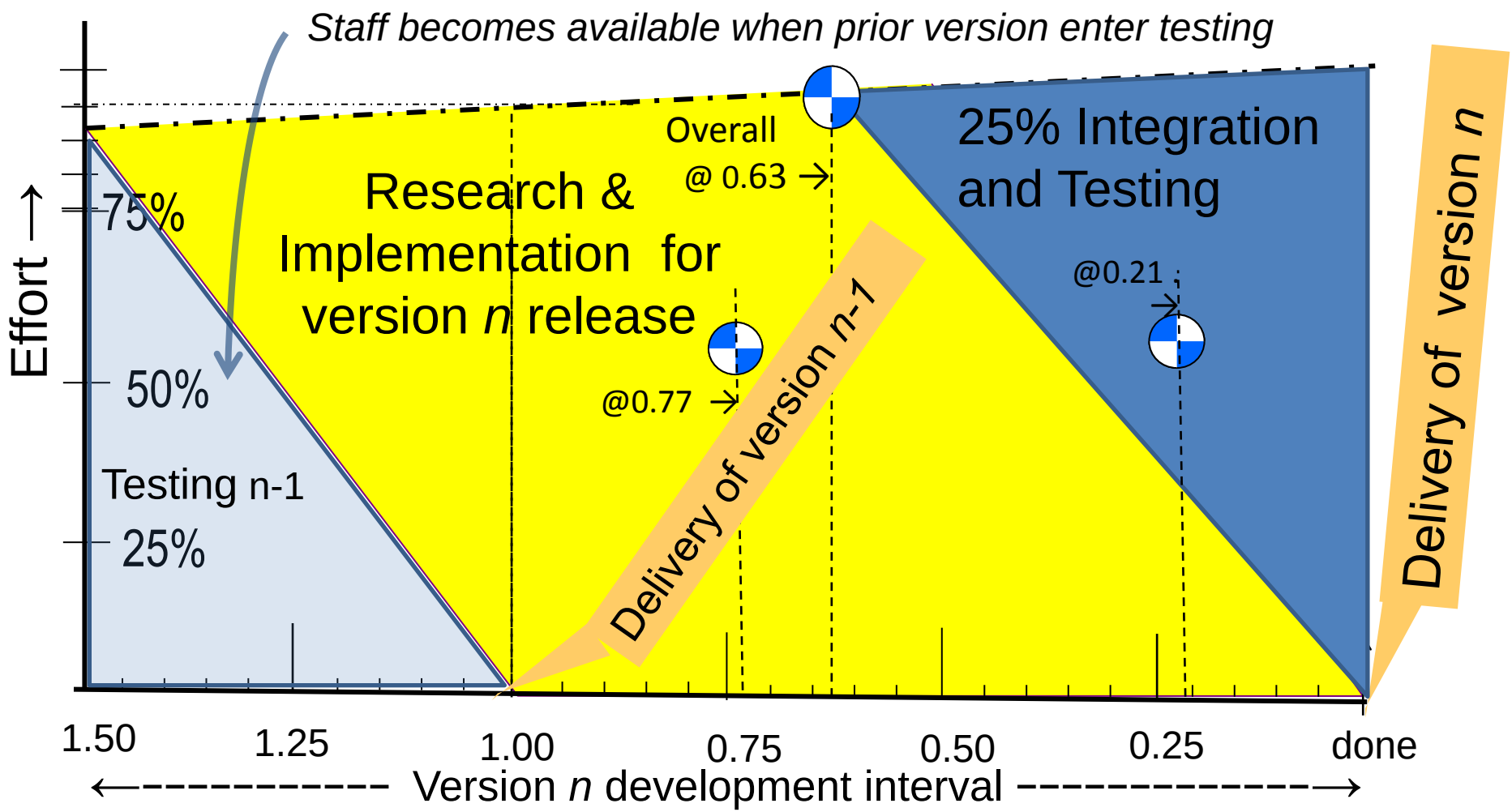
New considerations

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

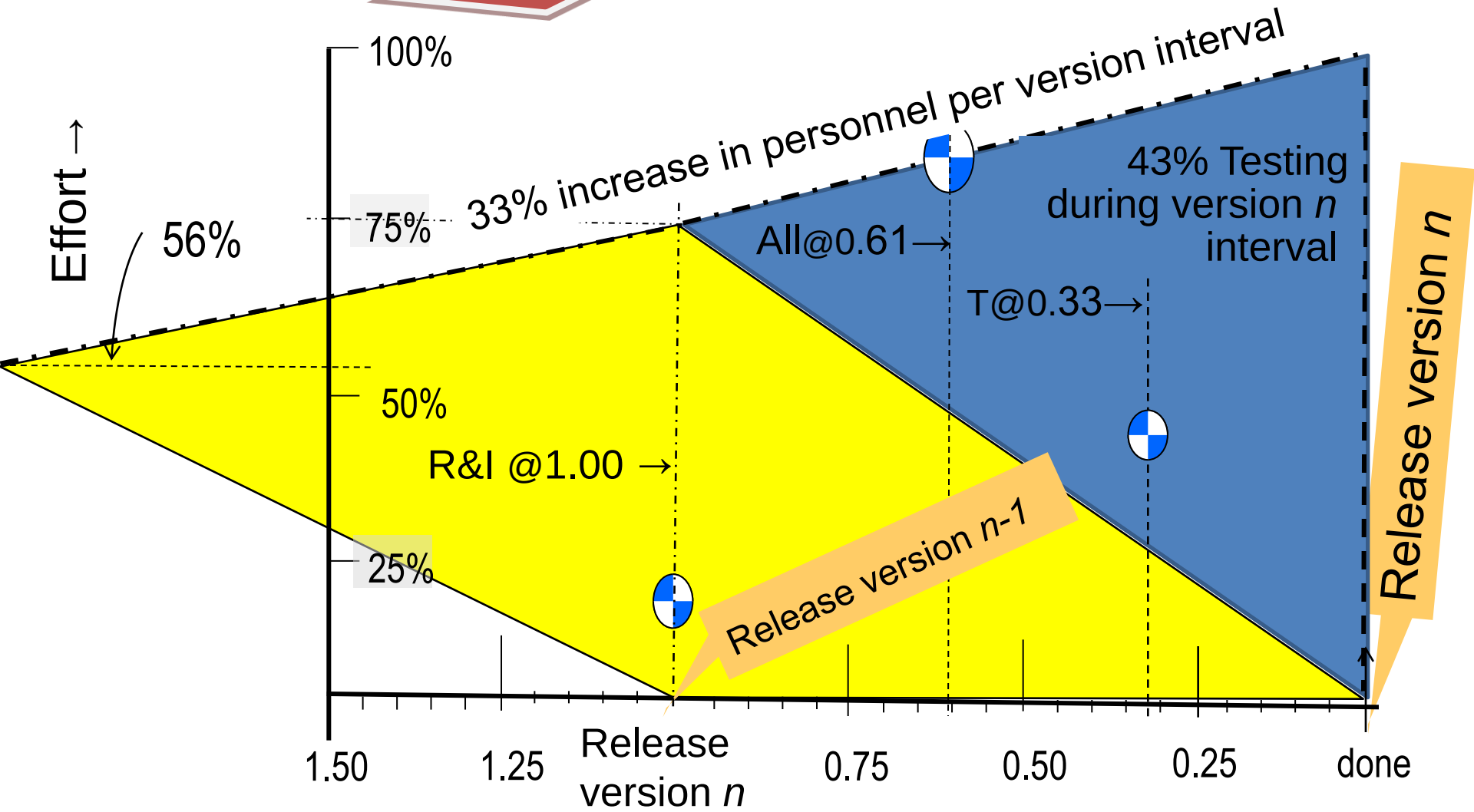
2nd version technical lag



Mature ongoing technical lag

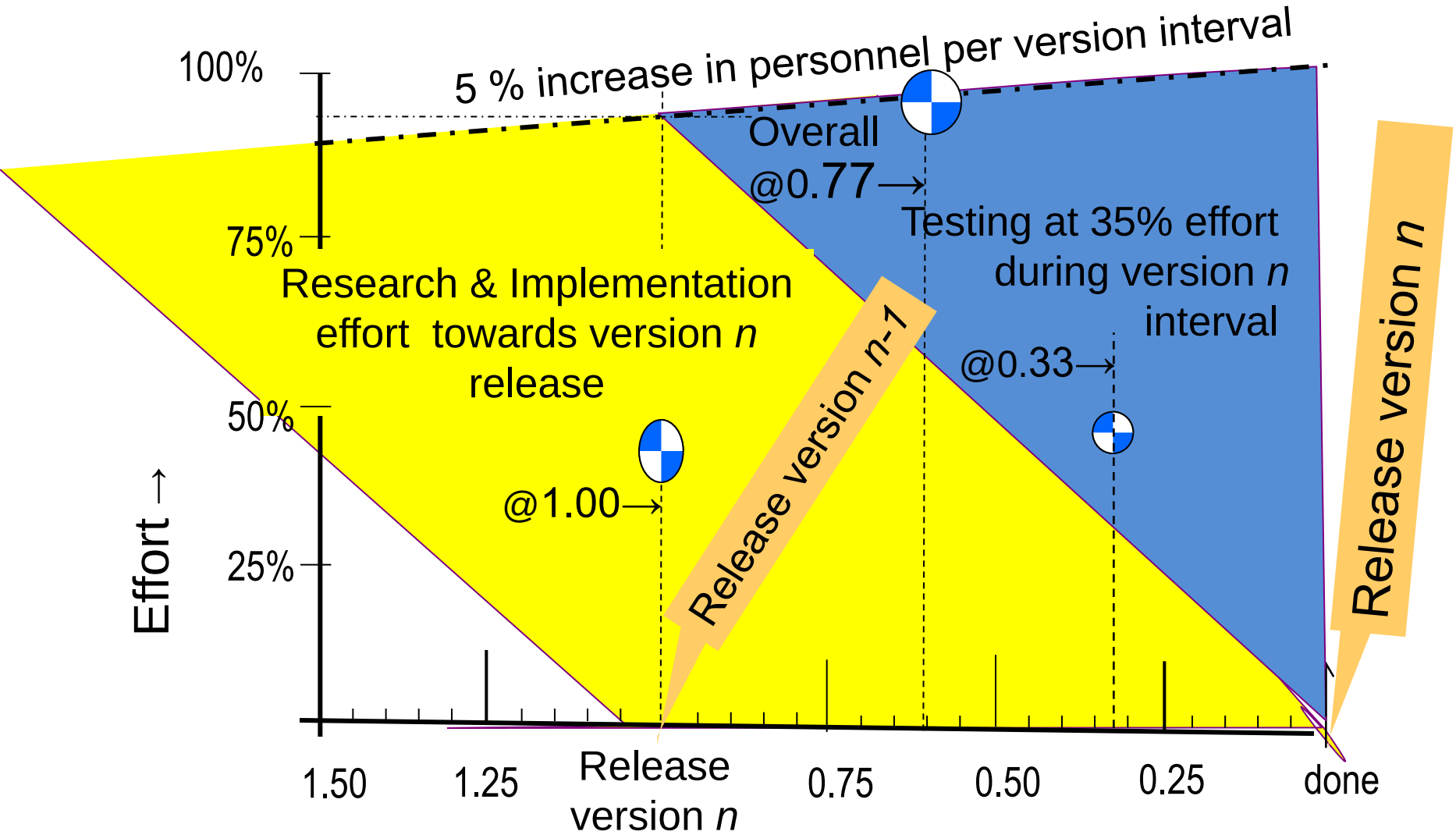


2nd Version substantial testing



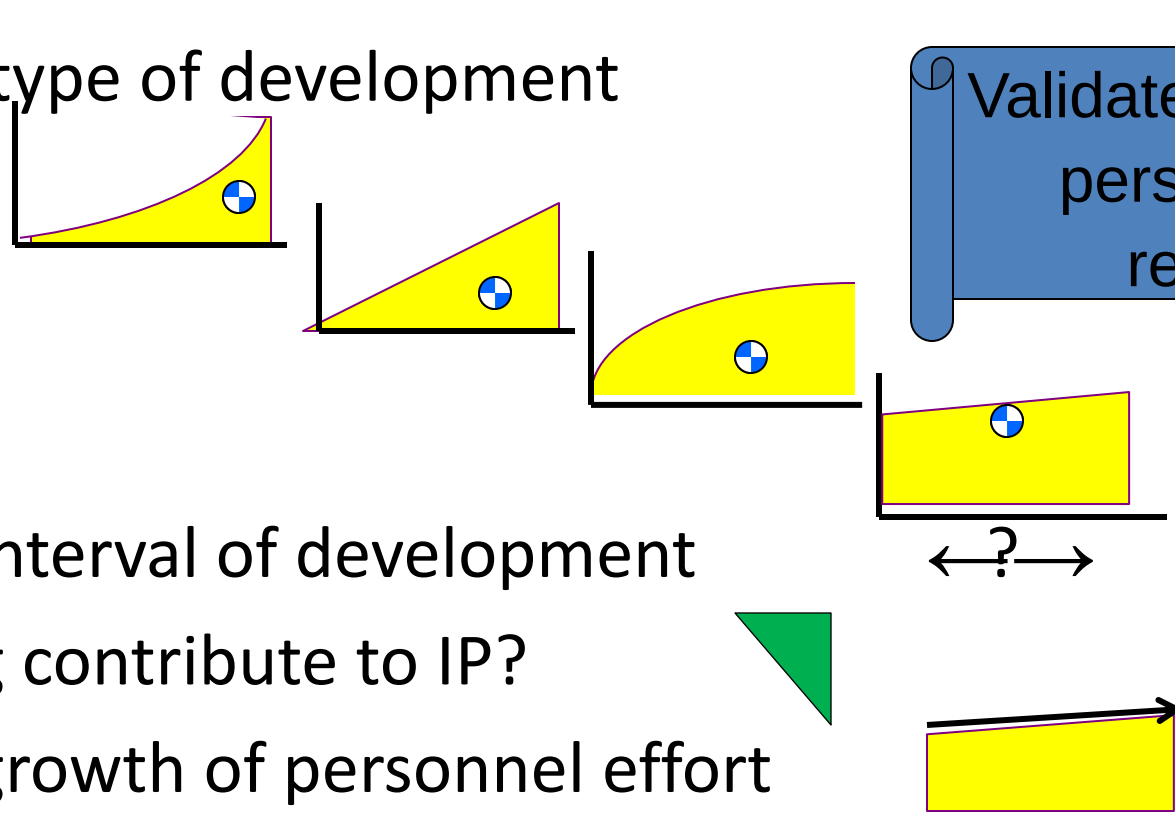
Version development,

mature growth, much testing



Determine technical lag

1. Determine type of development
   a. Startup
   b. Simple
   c. Mature
   d. Ongoing ?
2. Determine interval of development
3. Does testing contribute to IP?
4. Determine growth of personnel effort

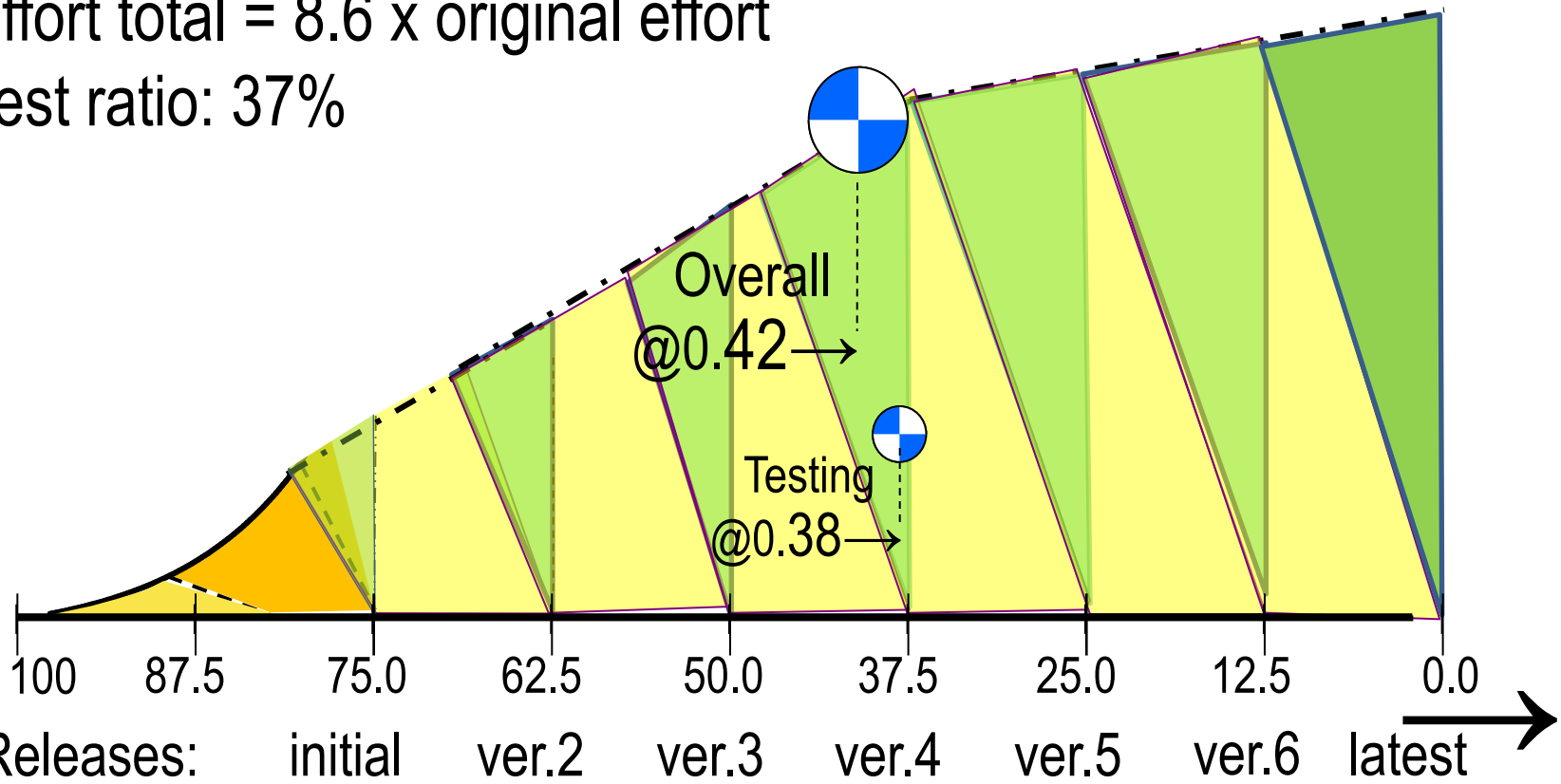


Lag, because it precedes IP diminution, has a large effect on economic valuation

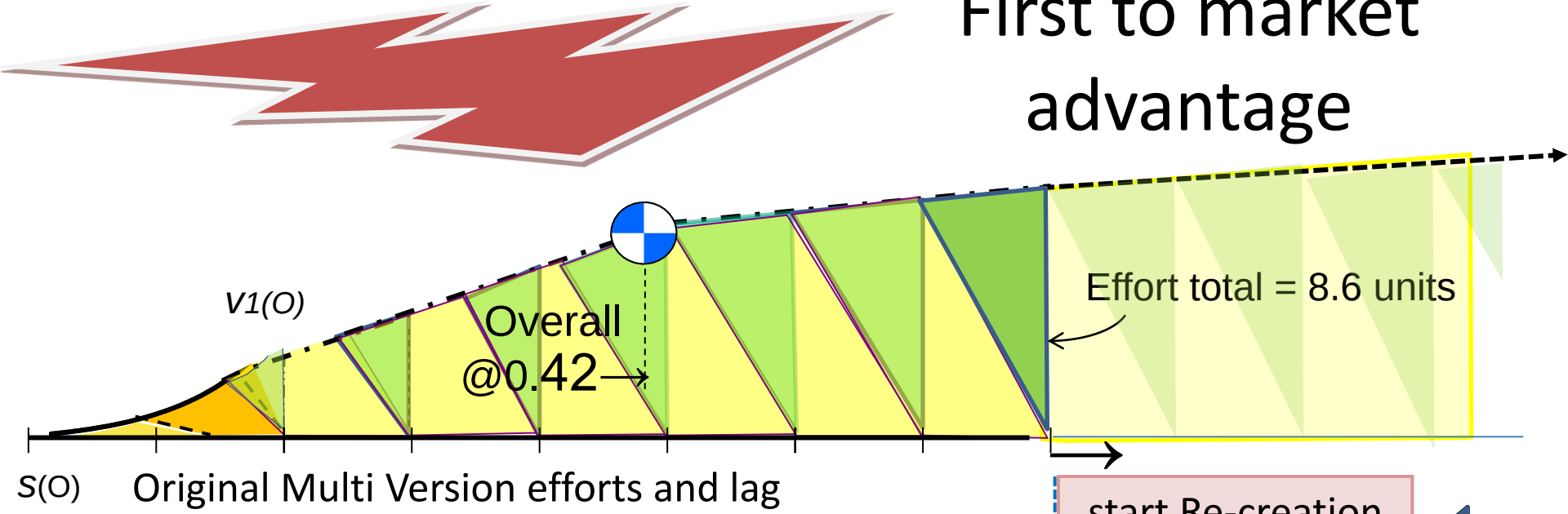
Multi Version product effort and lag

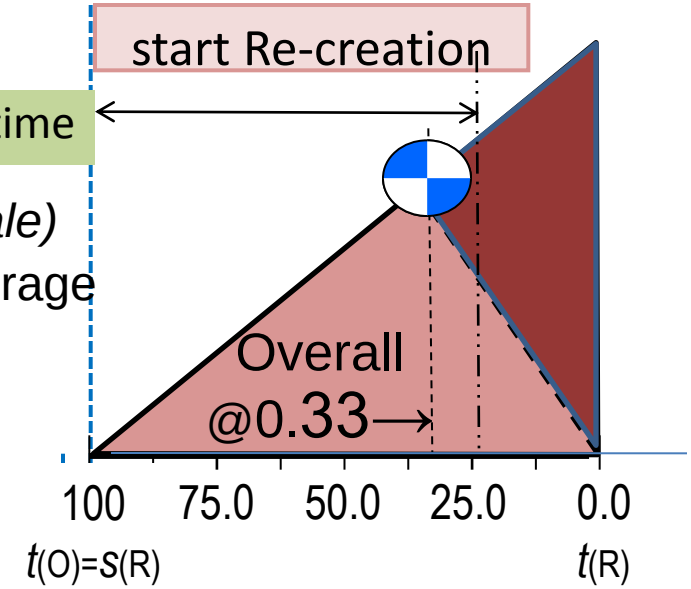
Effort total = 8.6 x original effort

Test ratio: 37%



First to market advantage





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
Effective Lag ratio = 0.23

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

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

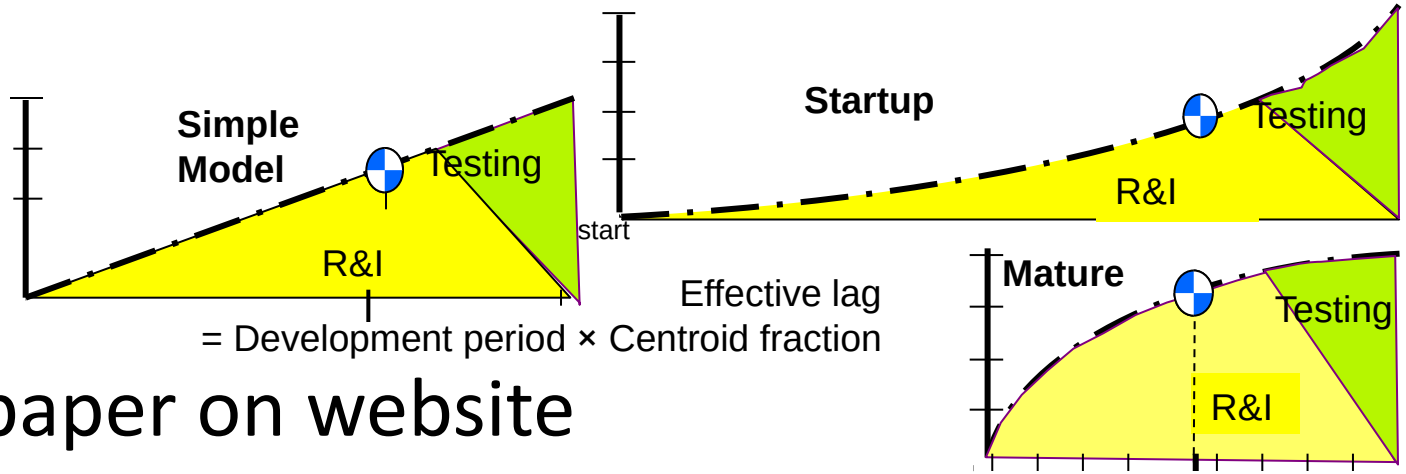
Lag conclusion
(snake in the grass)

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

Setting License fees

Say you want to delegate sales in Europe to some company EUsales that can do it easier over there

- How do you set the fees or royalties?
  1. You have computed a value of your SW of $1M
     - But without discounting, it is actually $1.6M = Σ(due old, slide 5)
     - You will also maintain the SW 1.36M = Σ(maintenance cost, slide 12)

     The total due is $3M
  2. You expect the European sales will be 40% of total, 20 000
     - The reason for not discounting is that funds arrive at the same times.
- To earn the same you should charge *1./2.*= $150/unit
  - It does not matter how EUsales sells it and what it charges
  - Complexities are required language, interface improvements