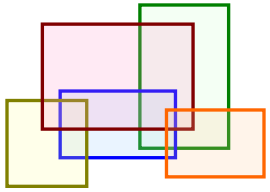
Requirements in the wild

How small companies do it

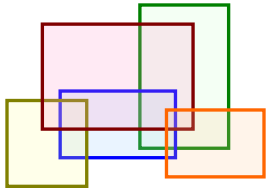
Jorge Aranda, Steve Easterbrook, and Greg Wilson
University of Toronto

RE'07, Delhi, India



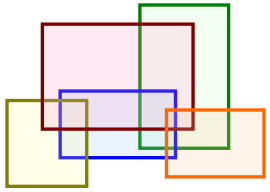
Why are we doing this?

- Small companies form a large part of the software industry
  - As an example, in the US in 2002:
    - 95% of all software firms have <50 employees
    - 21% of the total income of the field
    - 28% of all employees in the area
- And yet, *not a single paper* in the entire history of the RE conferences deals specifically with small companies
  - even though small companies are qualitatively different than their larger counterparts
- Anecdotal evidence told us that their practices differ significantly from those prescribed in the literature...
  - ...and that they haven't been much interested in what we have to say



Research questions

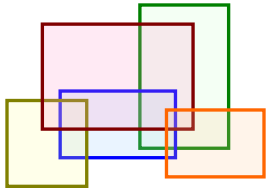
- How do small companies manage their requirements?
- How does the context of these companies affect them?
- Why do these companies adopt some practices and reject others?



Methodology

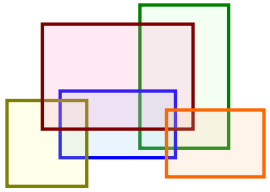
- Multiple-case exploratory case study
  - Exploratory studies: Gather data with the aim of deriving specific *hypotheses* for future study
    - Appropriate since we know so little about the domain
    - Multiple cases make for richer, more trustworthy hypotheses
  - Unit of analysis is a software *company*
    - Note: Not necessarily a software *team*

- Selection criteria
  - The company does software development as a primary activity
  - The company is small (<50 employees)
  - The company has been in operation for at least one year
  - (For convenience) the company must have offices in Toronto



Methodology (cont.)

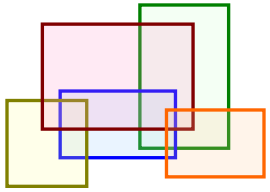
- Data collection through interviews and site visits
  - Interviewed partners, owners, or other persons holding leadership positions in each organization
  - 1-2 hour long interviews, 1-3 interviews per company
  - Open interviews covering a variety of requirements engineering issues, following our research questions
    - Elicitation, documentation, and communication of requirements
    - Forces affecting their requirements processes
    - Reasons for adoption/rejection of practices, processes, and tools
  - Non-judgmental listening stance
  - Find out what works for them, what doesn't, and why



The cases

	Endosymbiotic	Agilista	Spark	Bespoker	PhoneOffshore	Growing Web	Rentcraft
Company Size[1]	7	4	19	40-45	20-25	5	25
Longevity	15 months	13 years	5 years	5 years	7 years	3 years	12 years
Customers	Hospital	Manufacturing	News agencies & publishers	Banks & corporations	Telecoms	Varied (content management)	Rental companies
Type of offering[2]	Product, service	Projects	Product, service	Projects	Projects	Projects	Product
Project length /Release cycle	1 month	2 weeks	1 year	4 months – 2 years	~6 months	4 hours – 3 months	9 months – 1 year
Key requirements documents	Product backlog	Product backlog, user stories	*None*	Spec, development handbook	Statement of work, project plan	Cost worksheet, architecture & design	Analysis & est., product reqs' description
Signs of adaptation to niche	Co-location with customer	*Insufficient data*	Year-long negotiation processes	*Insufficient data*	Homegrown framework	Homegrown framework	*Insufficient data*
Cultural Cohesion	Previous company	Engineering	CS PhDs & MScs	Previous companies	Language & country	*None*	Previous companies
Analyst	Founder	Founder	CEO/CIO	Project lead	Project lead	Founder	Product manager
Mitigation of requirements errors	Monthly demos	Iterations	Iterations	Upfront analysis, iterations	Negotiation	*None apparent*	Upfront analysis, beta testing

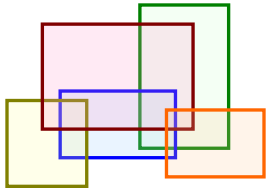
Notes: 1. Company sizes are approximate for cases where the company is currently recruiting and hiring new staff. 2. We categorized the company's activities according to where the requirements originate: "Projects" are custom development projects with a specific customer and limited duration, "Products" are applications intended for a wider market, and "Services" are long-term engagements (e.g web services).



Preliminary observations

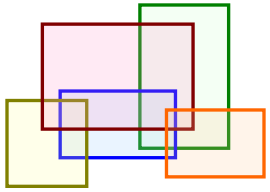
- A few notes before presenting our major findings:
  - All the companies we interviewed have requirements practices that work for them
    - Enough revenue to stay in business, and in most cases, to grow
  - They are all led by innovative and intelligent people
    - Generally knowledgeable about advanced software engineering concepts
    - Many years of experience in the software industry
  - *"These people don't know what they're doing"* doesn't cut it

Lesson 1:
Everyone does RE differently



Lesson 1:
Everyone does RE differently

- The diversity is striking
  - From detailed documentation to no documents whatsoever
  - From "planning it" to "correcting it"
  - From 4 hour to 2 year cycles
  - From sticking to a methodology to willingly dismissing all of them
- And yet, each considers that their choices are natural
- Several contextual variables appear to affect requirements practices:
  - Type of customers
  - Background and skill of developers
  - Preferences of founders
  - Nature of business environment
  - Spatial layout and geographical distance between offices
  - Number of employees



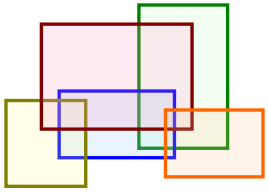
Lesson 1:
Everyone does RE differently

Hypothesis:
The diversity of RE practices in small companies can be explained as the result of evolutionary adaptation, as these companies have adapted to a specific niche.

- Software industry as eco-system
  - Differentiation occurs when companies adapt to fit a niche
  - Natural selection occurs when companies survive in a competitive environment by being better adapted to the niche than others

- Implications:
  - If the hypothesis is correct, *no generalized requirements technique will be suitable for all small companies*.
  - The value of any technique will vary significantly depending on the context of the company

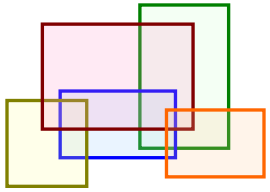
Lesson 2:
Strong cultural cohesion



Lesson 2:
Strong cultural cohesion

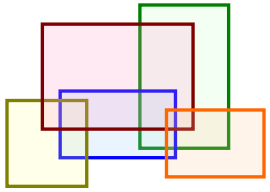
	Endosymbiotic	**Agilista**	**Spark**	**Bespoker**	**PhoneOffshore**	**Growing Web**	**Rentcraft**
Company Size[1]	7	4	19	40-45	20-25	5	25
Longevity	15 months	13 years	5 years	5 years	7 years	3 years	12 years
Customers	Hospital	Manufacturing	News agencies & publishers	Banks & corporations	Telecoms	Varied (content management)	Rental companies
Type of offering[2]	Product, service	Projects	Product, service	Projects	Projects	Projects	Product
Project length /Release cycle	1 month	2 weeks	1 year	4 months – 2 years	~6 months	4 hours – 3 months	9 months – 1 year
Key requirements documents	Product backlog	Product backlog, user stories	*None*	Spec, development handbook	Statement of work, project plan	Cost worksheet, architecture & design	Analysis & est., product reqs' description
Signs of adaptation to niche	Co-location with customer	*Insufficient data*	Year-long negotiation processes	*Insufficient data*	Homegrown framework	Homegrown framework	*Insufficient data*
Cultural Cohesion	Previous company	Engineering	CS PhDs & MScs	Previous companies	Language & country	*None*	Previous companies
Analyst	Founder	Founder	CEO/CIO	Project lead	Project lead	Founder	Product manager
Mitigation of requirements errors	Monthly demos	Iterations	Iterations	Upfront analysis, iterations	Negotiation	*None apparent*	Upfront analysis, beta testing

Notes: 1. Company sizes are approximate for cases where the company is currently recruiting and hiring new staff. 2. We categorized the company's activities according to where the requirements originate: "Projects" are custom development projects with a specific customer and limited duration, "Products" are applications intended for a wider market, and "Services" are long-term engagements (e.g web services).



Lesson 2:
Strong cultural cohesion

- In almost all cases, social characteristics shared by the group enabled it to simplify the tasks of requirements communication and coordination
  - *Homophily:* Natural attraction of individuals to others that have similar characteristics.
  - *Long term collaborations:* People "team up" for decades and across companies, achieving a deeper understanding of their partners' processes, work styles, and capabilities.
  - *Rejection of radical change:* Current requirements practices were negotiated, agreed, and settled in the past. Newcomers with radically different ideas are often received with hostility and do not last long.



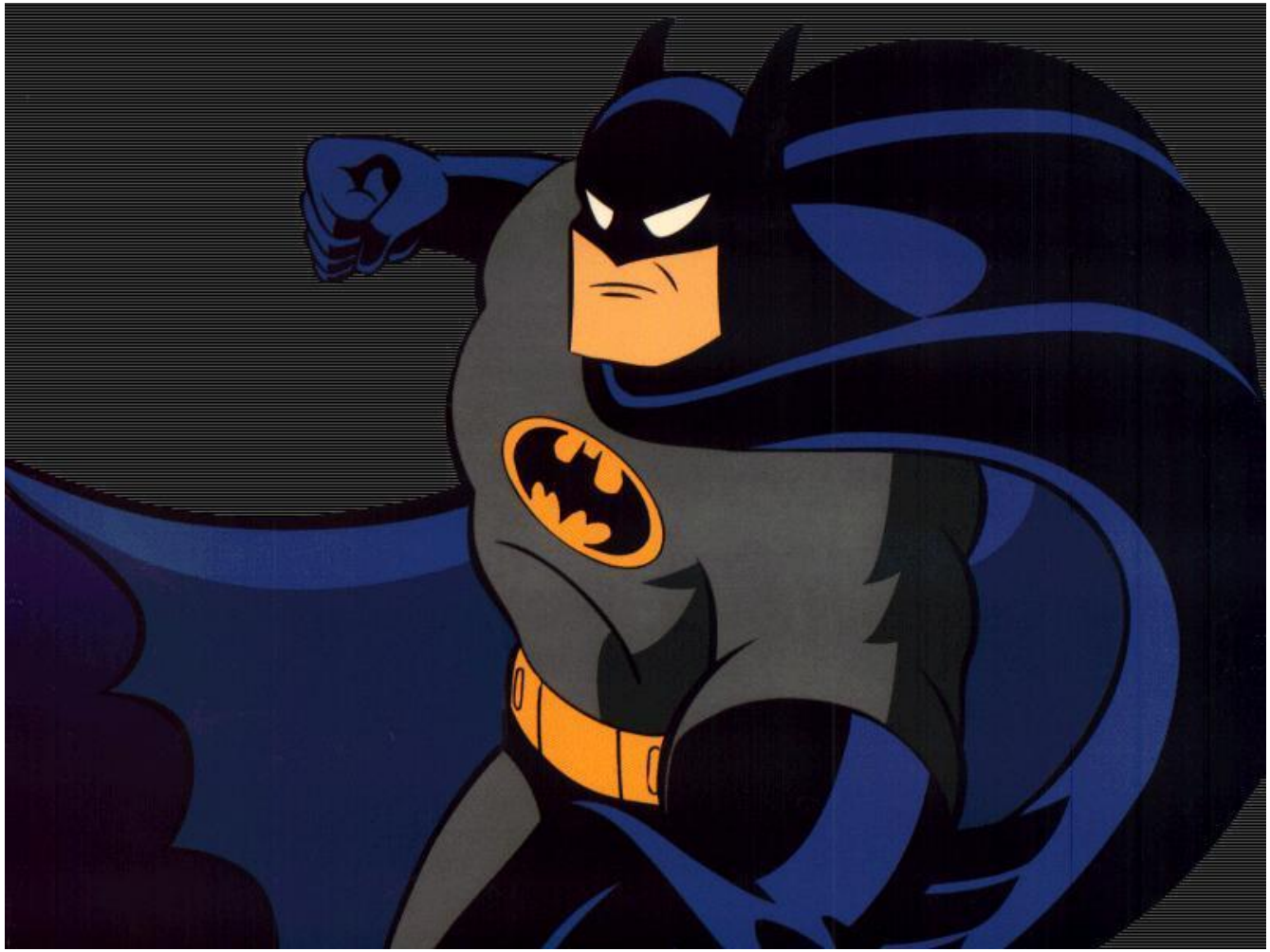
Lesson 2:
Strong cultural cohesion

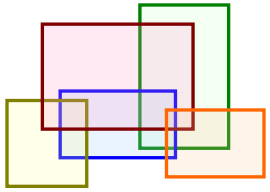
Hypothesis:
The choice of RE practices is irrelevant for small companies with strong cultural cohesion, as the efficiency of team dynamics overrides any benefits based on process.

*(Note that this hypothesis and the previous one are **competing** hypotheses)*

- Implications
  - We should be studying how teams acquire a shared understanding and a strong cohesion efficiently
    - Teams with strong cohesion don't need new requirements techniques or processes (they achieve shared understanding easily)
    - Teams without this cohesion might be able to overcome the problem through processes and documentation
  - Under this hypothesis, the diversity we observed is explained because, for these strongly cohesive companies, anything works

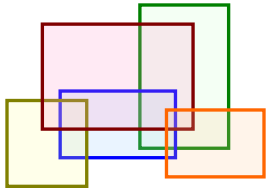
Lesson 3:
The CEO is the requirements engineer





Lesson 3:
The CEO is the requirements engineer

- For small company owners, requirements processes may well be one of the firm's most important activities
  - *They rarely give away the role of requirements engineer to their employees!*
    - In four of our seven cases, a founder or the CEO does the requirements work
    - In the other three, a trusted senior figure takes these responsibilities



Lesson 3:
The CEO is the requirements engineer

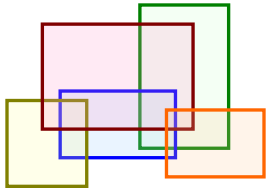
Hypothesis 1:
The skillset needed for successful requirements engineering is a subset of the skillset for successful entrepreneurship

- Most of our cases do not distinguish between the roles of "requirements engineer" and "customer liaison"
- The person eliciting requirements is often also the salesperson and contract negotiator, and needs skills matching these roles

Hypothesis 2:
Requirements engineering and business strategy are inseparable for small companies

*(Note that this hypothesis and the previous one are **complementary**)*

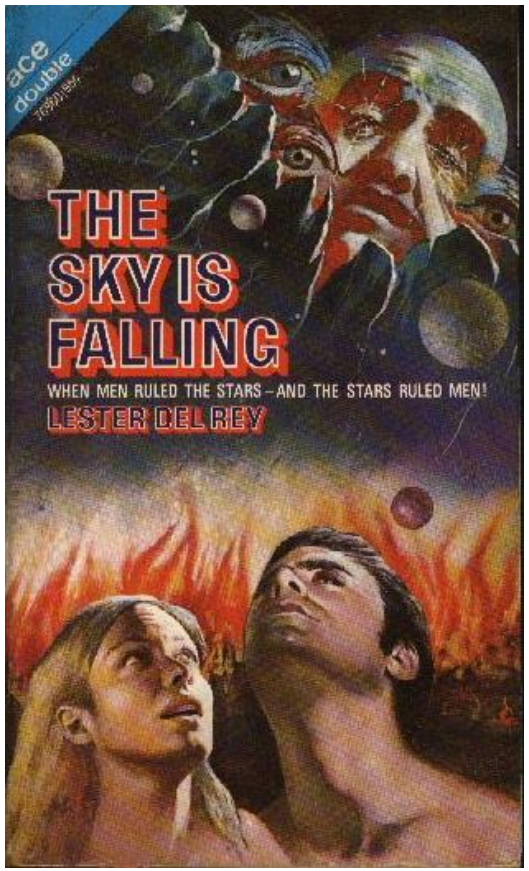
- To commit to a project implies locking a proportionally large amount of resources
- Requirements work is also strategic management work: the decisions of which projects to take and which features to include will define the company

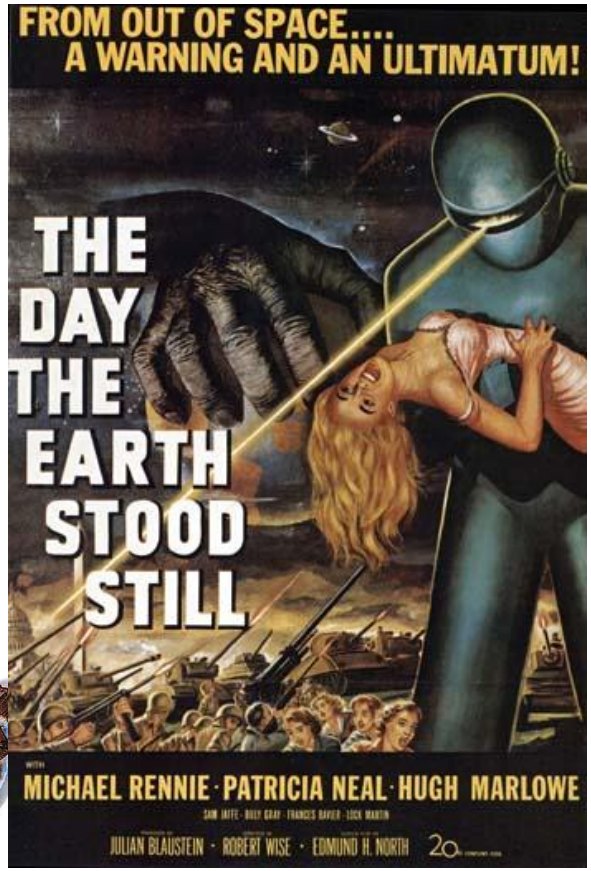


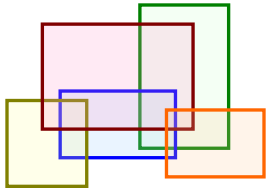
Lesson 3:
The CEO is the requirements engineer

- These explanations have important implications for our field
  - We often attempt to abstract the requirements process away from sales and strategic considerations
  - If this disconnect remains, it will be unlikely that owners of small companies find our proposals applicable to their situations

Lesson 4:
Requirements errors are not catastrophes



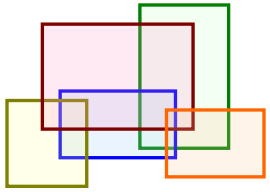




Lesson 4:
Requirements errors are not catastrophes

- Every person we interviewed had stories to share about requirements errors that compromised some of their projects...

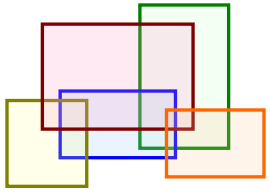
- ...and yet, nobody recalled any catastrophes caused by these errors
  - Sharp contrast with the commonly accepted perception of a "software crisis"
    - Especially of one caused by requirements problems



Lesson 4:
Requirements errors are not catastrophes

Hypothesis 1:
Small companies that survive their initial phase practice normal design*, which greatly decreases the risks associated with requirements engineering*

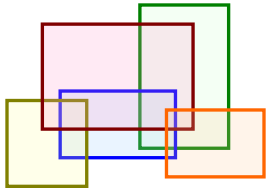
- These companies are well established, and appear to have adapted to their business niches
  - An important part of this adaptation may have been a shift from a *radical design* to a *normal design* approach to software development…
  - …allowing for the exploitation of skills and knowledge acquired previously, and decreasing risks dramatically



Lesson 4:
Requirements errors are not catastrophes

Hypothesis 2:
Small companies can fix their requirements problems more easily than large companies by virtue of being small

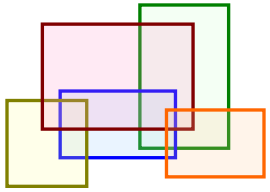
- Reduced communication and coordination overhead
  - It is easier to gather everyone and clear misunderstandings
  - Many of these companies share a (sometimes open) office space, enabling valuable information exchanges



Lesson 4:
Requirements errors are not catastrophes

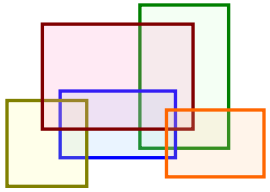
Hypothesis 3:
A single requirements catastrophe will drive a small company out of business

- Perhaps we did not observe companies with significant requirements problems because those went bankrupt already!
  - Internal validity bias



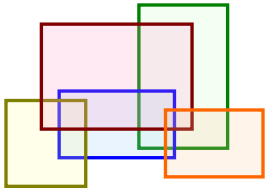
Lesson 4:
Requirements errors are not catastrophes

- Company owners do *not* perceive requirements errors as catastrophic
  - In most cases, requirements errors do not prompt them to take decisive actions to change their processes
  - Owners prefer to take the punches and maintain the processes that have kept them alive and growing, rather than to revolutionize and risk failure
- They will not adopt techniques that demand radical change!



Summary

- Diversity and adaptation *(everybody does RE differently)*
  - Understanding the context is essential
  - Proposed techniques may be helpful for some contexts, but not others
- Cultural cohesion
  - Process and documentation as remedies for weak cultural cohesion
  - Perhaps for teams with strong cohesion, any technique works
- CEO = Requirements Engineer
  - RE is also negotiation, salesmanship, and business strategy
  - They'll ignore us if we fail to incorporate these concerns
- The sky isn't falling
  - Our small companies are not desperate for a solution –what they already do, though imperfect, works for them
  - Incremental improvements favoured over radical changes



Questions?

Lesson 1: Everyone does RE differently

Hypothesis: *The diversity of RE practices in small companies can be explained as the result of evolutionary adaptation, as these companies have adapted to a specific niche*

Lesson 2: Strong cultural cohesion

Hypothesis: *The choice of RE practices is irrelevant for small companies with strong cultural cohesion, as the efficiency of team dynamics overrides any benefits based on process*

Lesson 3: The CEO is the requirements engineer

Hypothesis 1: *The skillset needed for successful requirements engineering is a subset of the skillset for successful entrepreneurship*

Hypothesis 2: *Requirements engineering and business strategy are inseparable for small companies*

Lesson 4: Requirements errors are not catastrophes

Hypothesis 1: *Small companies that survive their initial phase practice* normal design*, which greatly decreases the risks associated with requirements engineering*

Hypothesis 2: *Small companies can fix their requirements problems more easily than large companies by virtue of being small*

Hypothesis 3: *A single requirements catastrophe will drive a small company out of business*

Recommendations:

State the context
Connect RE research to business and social concerns
Provide the evidence
Provide incremental improvements

Jorge Aranda, Steve Easterbrook, and Greg Wilson
University of Toronto
{jaranda, sme, gvwilson}@cs.toronto.edu