Core concepts

Top warnings in this book

Top warning	Page number	Section
Using inheritance may sometimes introduce tight coupling that limits the flexibility and possibility to evolve our system.	38	2.5.2
The exception types thrown by your API are part of your contract. Beware leaking them.	55	3.4
Designing highly extensible code often increases the overall complexity of our solution.	94	4.5
Using an API (stream) without a complete understanding of its methods may lead to substantial performance degradation of our processing.	98	5.1.1
The lack of downstream components encapsulation allows us to deliver quicker at the beginning but limits the possibilities of evolution in the long term.	139	6.3.1
The idea that you can just store everything as UTC, and you'll never have time zone issues is a common myth. Storing UTC is fine in many cases, particularly timestamps, but loses important data in other cases. Don’t accept it as a silver bullet.	197	7.4.4
Our data partitioning may impact the possible ways of how we use the data. In the most serious cases, it may make some big data processing logic impossible to implement.	215	8.3.2
Some of the unchanged defaults of the library that you use can critically impact your application.	232	9.1
Some tools that work correctly in a one-node context may break its correctness in the multi-node environment.	270	10.3.1
Even if one of the systems in the pipeline offers effectively exactly-once, if our processing involves *N* remote system calls, those guarantees will not be held.	300	11.4.3
Breaking changes in a library incur cost throughout a community, even when properly versioned using Semantic Versioning. Consider the impact on consumers of the library—both direct and indirect.	319	12.2.2
Not all functional patterns should be used in a language that provides functional programming patterns, but is not a functional language from the ground up.	369	13.3.1