1) Discuss the prototyping model. What is the effect of designing a prototype on the overall cost of the project.

Ans:- Prototype is the act of making the software applications prototypes which is basically an incomplete version of the software program that is being developed. It takes place in software development and is comparable to prototyping as known in other fields like that of manufacturing and mechanical engineering. However, it is completely different from that of the final product and stimulates only a few aspects.

It depends on how you define “cost”. If you define cost in the traditional sense of the word - i.e. cost of labor, time and effort - then the cost will increase as your adding extra activities. Especially since these activities are mainly meant to learn and not to necessarily generate tangible out, things like prototyping will add additional cost.

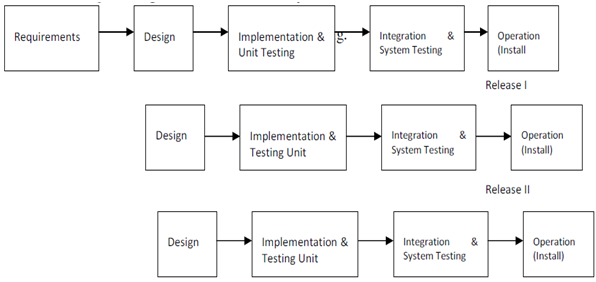
However, if you define cost in terms of cost of delay or opportunity cost, then discovery through prototypes will reduce overall cost. For me, the key about using prototypes - irrespective of level of fidelity - is the learning about potential risks:

1. **Customer risk**- Will the user buy this (or choose to use this)? Why (not)?
2. **Usability risk** - Can the user figure out how to use this? What works, what doesn’t (and why)?
3. **Implementation risk** - Can our engineer build this? Is it technically viable?
4. **Business risk** - Can our stakeholders support this? Is it viable from a business perspective?

The cost of not identifying and mitigating these risks early and often can be significant, if you think of cost in terms of missed opportunities and delays in speed to market.

2) Compare iterative enhancement model and evolutionary process model.

Ans:- Iterative Enhancement Model: This model has the similar phases as the waterfall model, but with fewer restrictions. In general the phases occur in the same order as in the waterfall model but these may be conducted in several cycles. A utilizable product is released at the end of the each cycle with each release providing additional functionality.



Evolutionary Development Model: Evolutionary development model bear a resemblance to iterative enhancement model. The similar phases as defined for the waterfall model occur here in a cyclical fashion. This model is different from iterative enhancement model in the sense that this doesn't require a useable product at the end of each cycle. In evolutionary development requirements are implemented by category rather than by priority.

3) As we move outward along with process flow path of the spiral model, what can we say about software that is being developed or maintained.

Ans:- Each phase of the Spiral Model is divided into four quadrants as shown in the above figure. The functions of these four quadrants are discussed below-

1. **Objectives determination and identify alternative solutions:** Requirements are gathered from the customers and the objectives are identified, elaborated, and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.
2. **Identify and resolve Risks:** During the second quadrant, all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and the risks are resolved using the best possible strategy. At the end of this quadrant, the Prototype is built for the best possible solution.
3. **Develop next version of the Product:** During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.
4. **Review and plan for the next Phase:** In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.

**Risk Handling in Spiral Model**  
A risk is any adverse situation that might affect the successful completion of a software project. The most important feature of the spiral model is handling these unknown risks after the project has started. Such risk resolutions are easier done by developing a prototype. The spiral model supports coping up with risks by providing the scope to build a prototype at every phase of the software development.

4) Explain the Scrum Agile methodology.

Ans:- Agile scrum methodology is the combination of the agile philosophy and the scrum framework. Agile means “incremental, allowing teams to develop projects in small increments. Scrum is one of the many types of agile methodology, known for breaking projects down into sizable chunks called “sprints.” Agile scrum methodology is good for businesses that need to finish specific projects quickly.

The Agile Manifesto also enacted 12 principles in reference to software development and was later reconfigured to fit a wider perspective of users:

1. Customer satisfaction
2. Early and continuous delivery
3. Embrace change
4. Frequent delivery
5. Collaboration of businesses and developers
6. Motivated individuals
7. Face-to-face conversation
8. Functional products
9. Technical excellence
10. Simplicity
11. Self-organized teams
12. Regulation, reflection and adjustment

Scrum is a framework for effective collaborations among teams working on complex products. Scrum is a type of agile technology that consists of meetings, roles, and tools to help teams working on complex projects collaborate and better structure and manage their workload. Although it is most often used by software development teams, scrum can be beneficial to any team working toward a common goal.

These are some of the collective benefits of agile scrum methodology:

* [Flexibility and adaptability](https://www.businessnewsdaily.com/5696-pivot-business-strategy-shift.html)
* Creativity and innovation
* Lower costs
* Quality improvement
* Organizational synergy
* Employee satisfaction
* Customer satisfaction

The greatest benefit of agile scrum methodology is its flexibility. With the sprint-based model, the scrum team typically receives feedback from stakeholders after each sprint. If there are any problems or changes, the scrum team can easily and quickly adjust product goals during future sprints to provide more valuable iterations. This way, stakeholders are happier because they get exactly what they want after being involved every step of the way.

Compare this with traditional project management systems, in which stakeholders do not provide frequent feedback and time is wasted making changes to the product halfway through development – or worse, such as the teams needing to start from scratch after the product has already been built.

To implement agile scrum methodology, there must be either a scrum expert in the company or an outside consultant to ensure scrum principles are being applied correctly. Agile scrum methodology involves precise execution and could result in serious problems if not done properly.

5) Explain the utility of Kanban CFD reports.

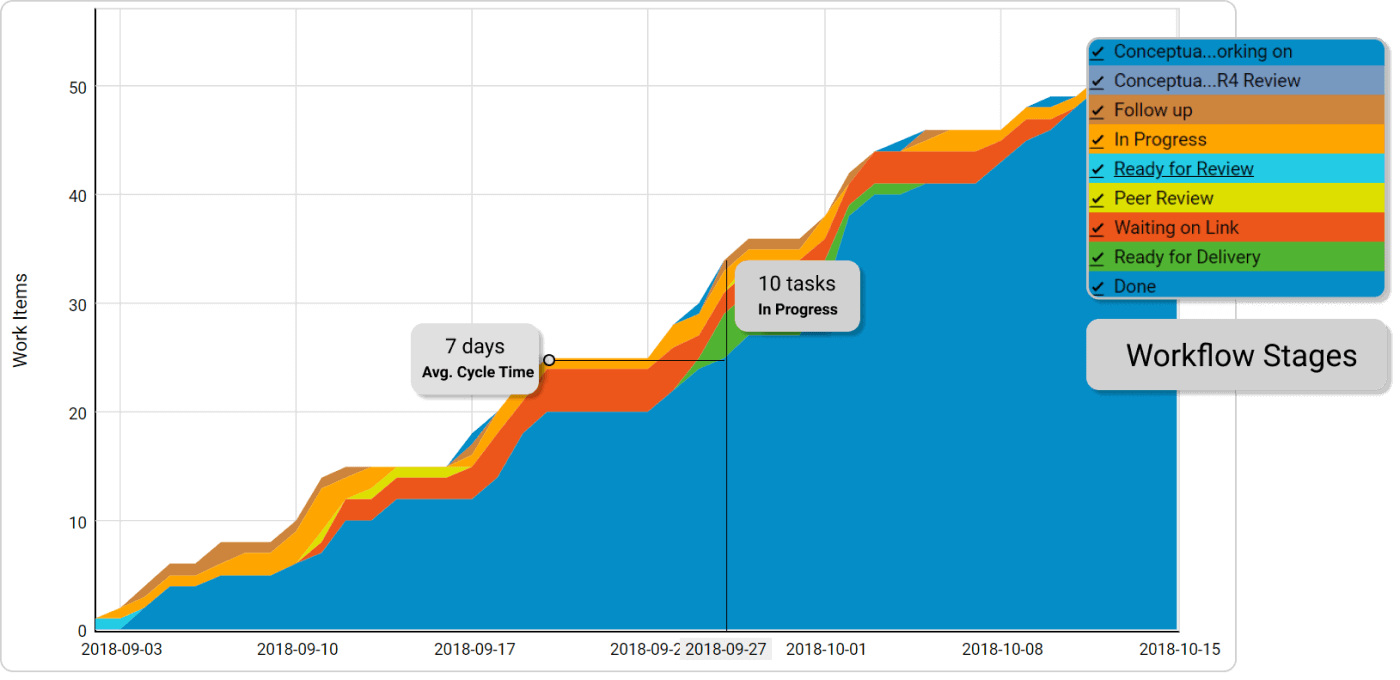
Ans:- The cumulative flow diagram (also known as CFD) is one of the most advanced Kanban and Agile analytics charts. It provides a concise visualization of the three most [important metrics of your flow](https://kanbanize.com/agile/project-management/agile-metrics):

* Cycle time
* Throughput
* Work in progress

Its main purpose is to show you how stable your flow is and help you understand where you need to focus on making your process more predictable. It gives you quantitative and qualitative insight into past and existing problems and can visualize massive amounts of data.

**How To Read a Cumulative Flow Diagram**

The chart tracks the total number of work items in the columns of the "In Progress" section on your Kanban board each day.



The horizontal axis of the CFD represents the time frame for which the chart is visualizing data. The vertical axis shows the cumulative number of cards in the workflow at various points in time.

The differently colored bands that divide sections of the upward flow are the different stages of your workflow as they appear on the Kanban board itself. The bands always go up or sideways in accordance with the number of assignments that go through your process.

The top line of each band on the cumulative flow chart represents the entry point of tasks in the respective stage of your Kanban board, while the bottom one shows when it leaves it. If a line becomes flat, nothing arrives in the corresponding stage, or nothing is leaving it.

Using a CFD, you can get an idea of how long your tasks' approximate cycle time is.

This is possible by measuring the horizontal distance between the top line of the first stage on the cumulative flow diagram and the bottom line of the last "in progress" stage.

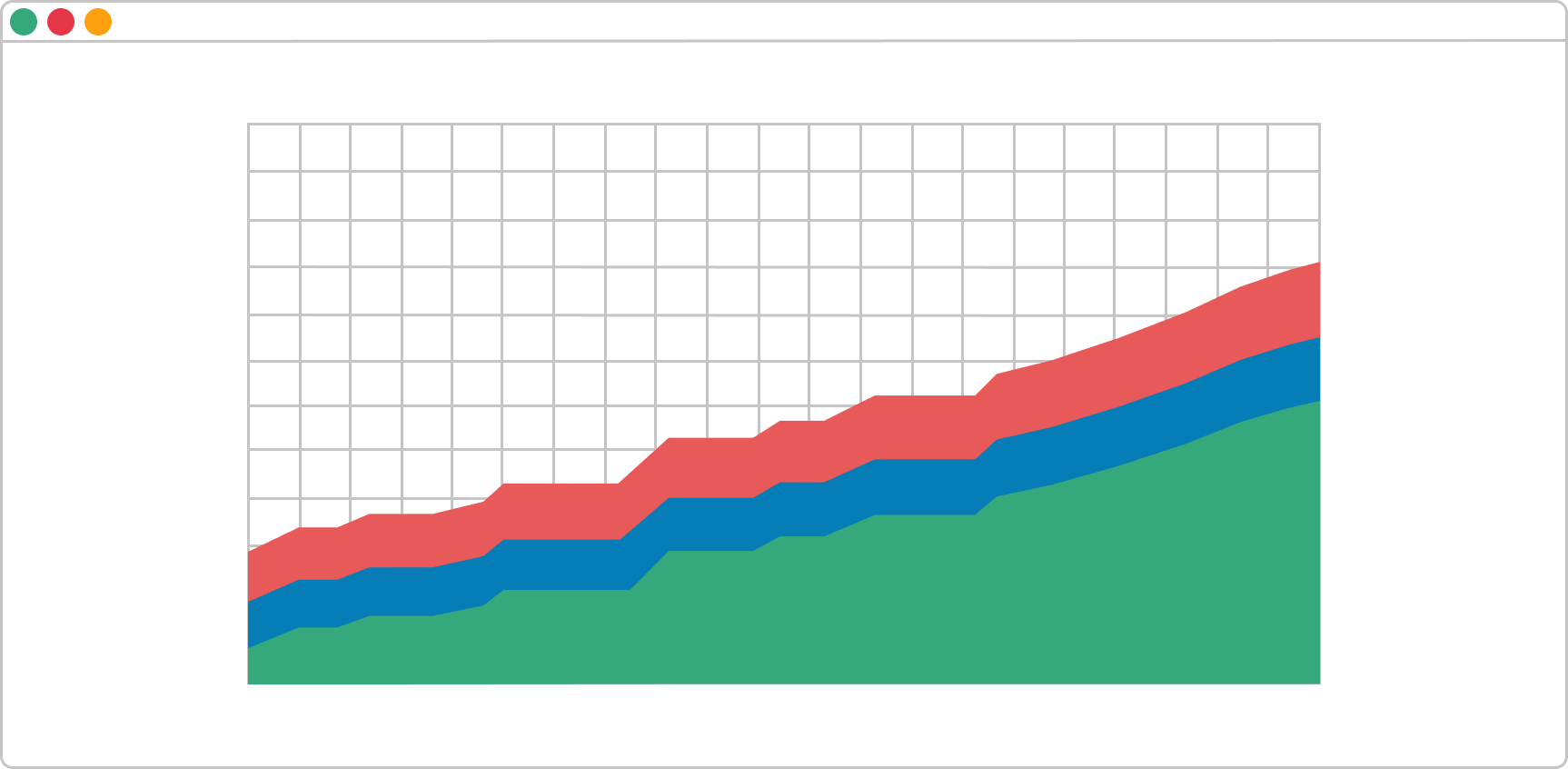
The number of days/weeks/months that have passed is the approximate average cycle time of your team’s assignments for the time frame.

***The distance between the lines of a CFD will show you the problems of your workflow.***

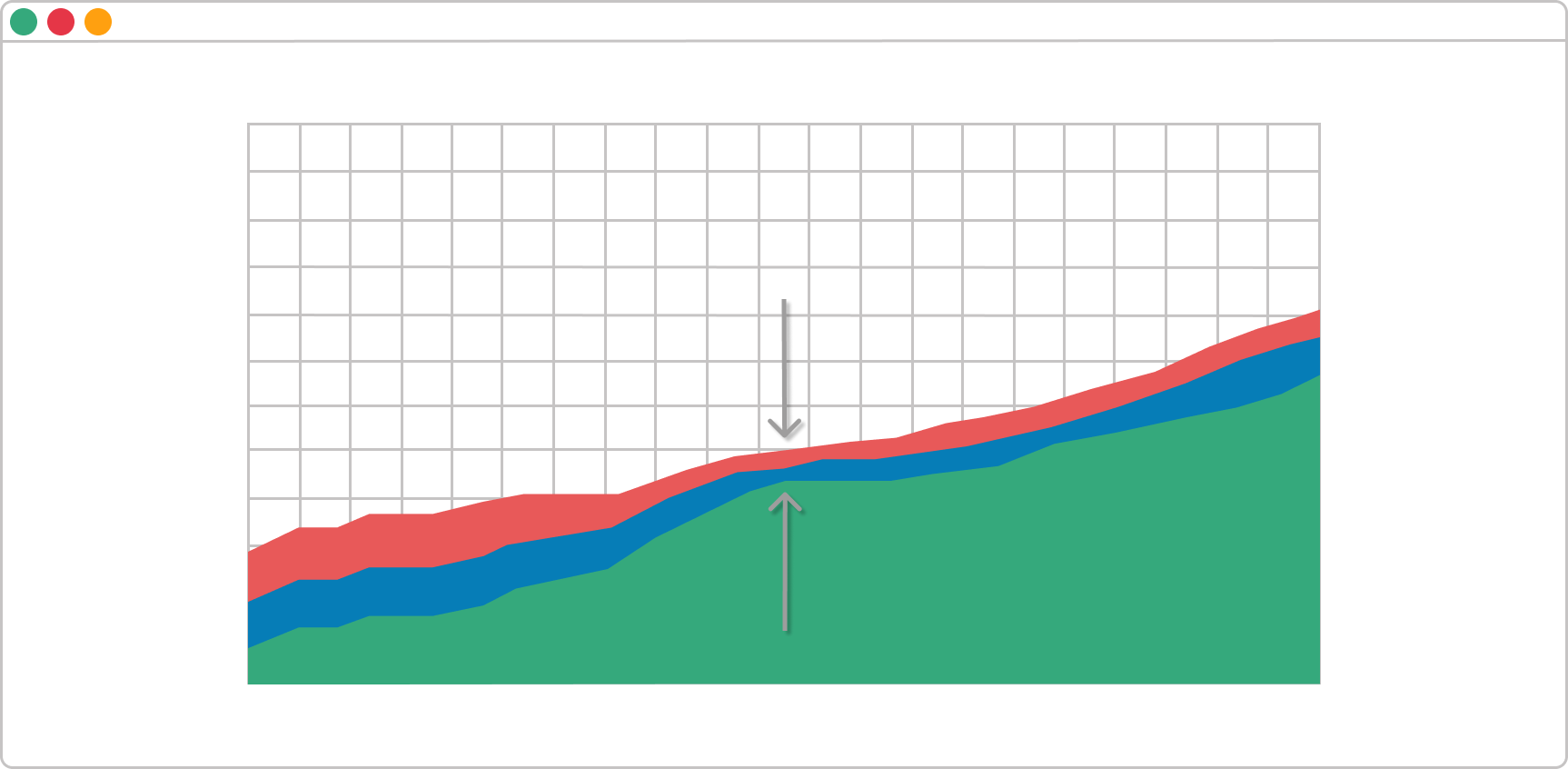
**Understanding the data on a CFD Chart**

You can spot whether your process is stable in just a single glance by looking at how the top and the bottom line of each band in your cumulative flow diagram are progressing.

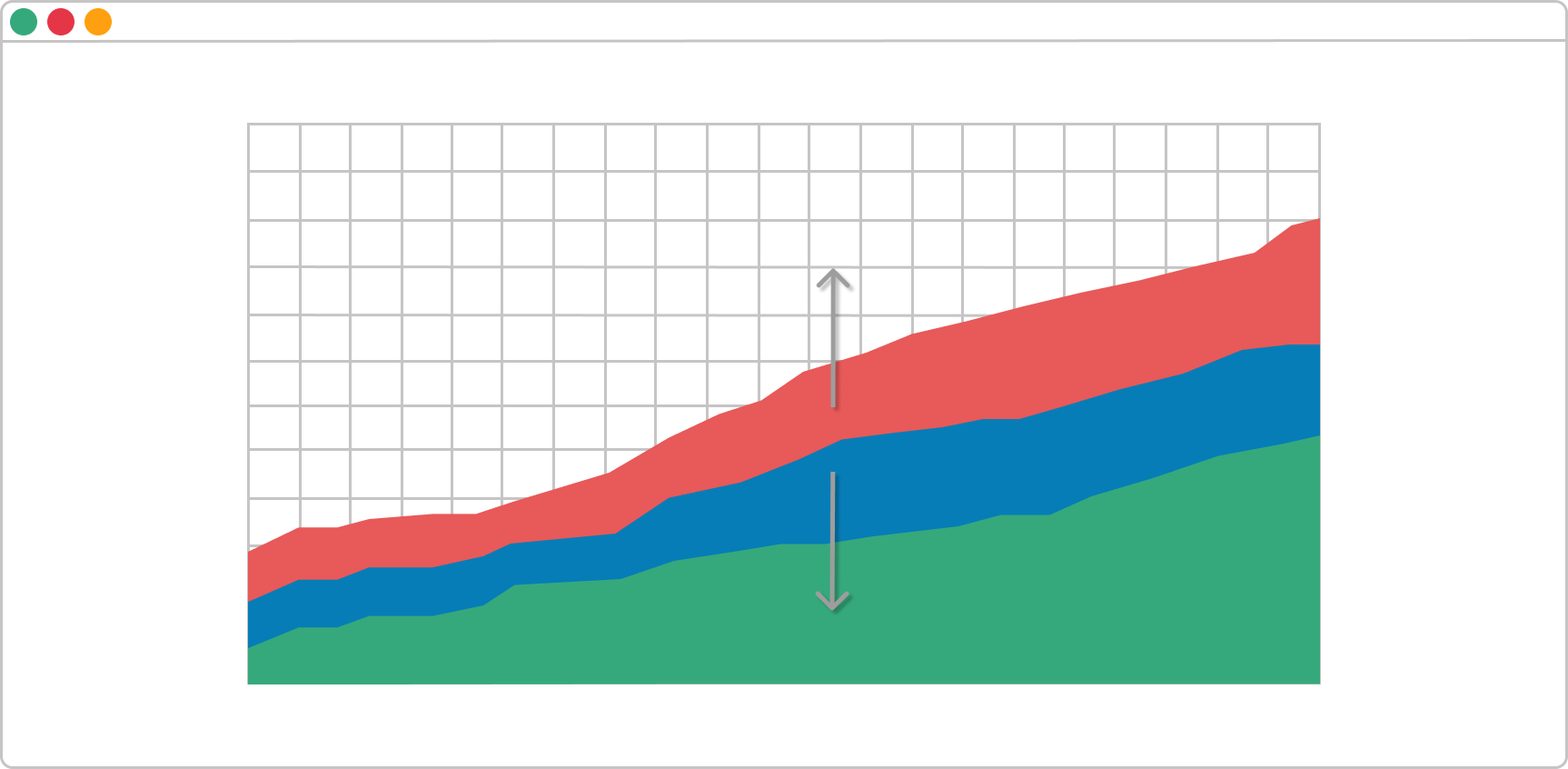
There are three common scenarios:

**The Bands are Progressing in Parallel**

This means that your throughput is stable, and new tasks are entering your workflow in parallel to those that are leaving it. This is the ideal outcome and shows that you can focus your efforts on shortening your assignments' cycle times.

**A Band is Rapidly Narrowing**

If a band on your CFD is continuously narrowing, that means that the throughput of the stage it represents is higher than the entry rate. This is a sign that you’ve got more capacity than you really need at this stage, and you should relocate it to optimize the flow.

**A Band is Rapidly Widening**

Whenever this happens on a cumulative flow diagram, the number of cards that enter the corresponding stage on the Kanban board is higher than the number of assignments leaving it. It is a common problem caused by multitasking and other waste activities that don’t generate value.

There are many possible actions to resolve this issue. However, if this is not generated by a dependency on external stakeholders, you should reconsider the existing [WIP limits](https://kanbanize.com/kanban-resources/getting-started/what-is-wip/) on your Kanban board and focus on finishing tasks that are in progress before starting new ones.

You should be aware of a possible 4th scenario as well – the bands are going down. **If any band on your cumulative flow diagram goes down, the diagram is incorrect.**A task should never disappear from your workflow.

In conclusion, the cumulative flow diagram is an advanced analytic tool that will give you an accurate picture of how stable your process is and how efficient your team is. You can get a piece of actionable advice on where you need to focus your efforts in order to improve your process in a single glance.