While it is highly-possible that the people who wrote the ACM Code of Ethics read Parnas's "Rational Design Process" paper, I do not believe that the paper contributes to the code. My table-mates Vincent Sordo and Natalia Tupy both believe that his paper did contribute to the Code though, so I cannot say for sure. But the Code does address Parnas's paper, specifically with sections 3.6, 2.1 and 1.3.

Section 3.6 of the ACM Code of Ethics states that as a software professional, you should "create opportunities for members of the organization to learn the principles and limitations of computer systems." This is a central part of Parnas's paper, because while he claims that requirements documents are near impossible because both the customer and the professional can not see everything needed to build a system from the beginning, he does want the professional to continuously improve the document, and write down the problems they encountered to make it look like they knew everything from the beginning. If the customer looks at how the document changed from the beginning to end they can see some of the limitations in computers and software design. With the professionals help, they can become a better customer by looking at the types of problems a software developer runs into while implementing a specification.

Section 2.1 of the ACM Code of Ethics states that as a software professional, you should "strive to achieve the highest quality, effectiveness and dignity in both the process and products of professional work." Before class today, I would argue that Parnas's paper does not promote this at all. Parnas essentially wants us to cover up our bad work with a high quality document and pretend that we did good work. But, I have changed my opinion thanks to opinions expressed in class. The point the Parnas is really trying to make is that humans are fallible. They do not know everything, and in the beginning of a design, they know even less. But as time goes on they have a duty to create as best as they can and update their initial documentation to show how the system is now, and why it is of the highest quality.

Section 1.3 of the ACM Code of Ethics states that as a software professional, you should "be honest and trustworthy." This could be looked at in two ways. First off, Parnas is essentially telling us

to lie, by changing a document that the customer and the professional agreed on as the requirements of a project. But in reality, Parnas is leading us to something much more important, he wants us to return a document to the customer that explains exactly what we have created does, and how it would have been defined initially if the two of them had thought of everything initially.

I feel that Cal Poly's computer science department promotes all three of these sections, but of these three, it promotes section 2.1 the most. In all of our classes we, as students, are asked to either follow a set of requirements, or define a set of requirements and then follow them. But our teachers also encourage us to show them that their specifications are broken or incorrect. In fact, I remember a scene in my 103 class where Dr. John Dalbey changed an entire assignment when it was due in three days. Students were in uproar, because many had spent at least a week implementing his previous specification. His statement was simple though. It had been brought to his attention by a classmate of ours that the data structure we were building had a bug, and would go into an infinite loop no matter how you programmed it. The spec was designed to fail. Dr. Dalbey changed the spec because now, not only was it easier to implement, but it made more sense. We as students learned a lot that day, and future classes benefited by our ability to teach the teacher and promote our highest quality of work.